Name: Himesh Baraik

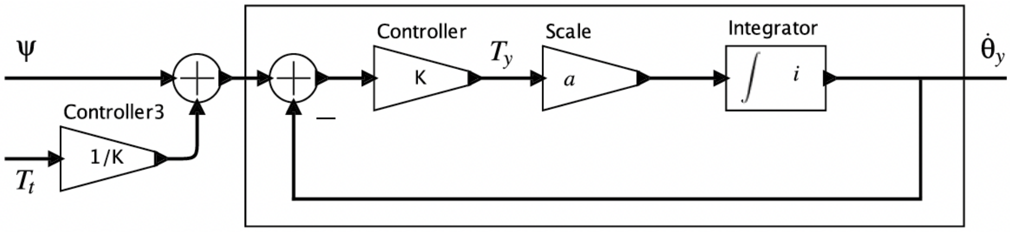
Roll No: 210447 (e.g. 170001)   Dept.: EE (e.g. CSE)

Indian Institute of Technology Kanpur
CS637 Embedded and Cyber-Physical Systems
Homework Assignment 1
Deadline: September 13, 2024

Total: **60 marks**

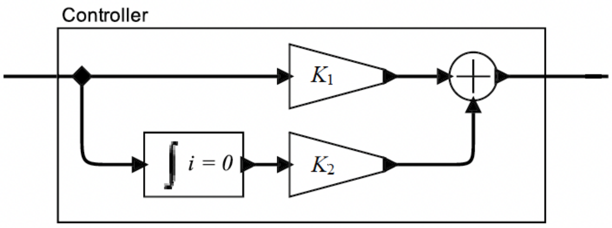
Problem 1. (20 points)

(a) Construct a Simulink model of the helicopter control system shown in the figure below.



Choose some reasonable parameters and plot the actual angular velocity as a function of time, assuming that the desired angular velocity is zero, $\psi(t) = 0$, and that the top-rotor torque is non-zero, $T_t(t) = bu(t)$. Give your plot for three different values of K and discuss how the behavior varies with K.

(b) Modify the model of part (a) to replace the Controller (the simple scale-by-K actor) with the alternative controller shown in the figure below.



This alternative controller is called a proportional-integrator (PI) controller. It has two parameter K_1 and K_2. Experiment with the values of these parameters, give some plots of the behavior with the same inputs as in part (a), and discuss the behavior of this controller in contrast to the one of part (a).

Solution:

Let us take the parameters of the system to be to be:
1. Initial angular velocity (i in the integrator block) = 4
2. Scale, a = 2

Controller3: This transformation simply relies on the algebraic fact that for any real numbers a1, a2, K, Ka1 + a2 = K(a1 + a2/K)

Name: Himesh Baraik

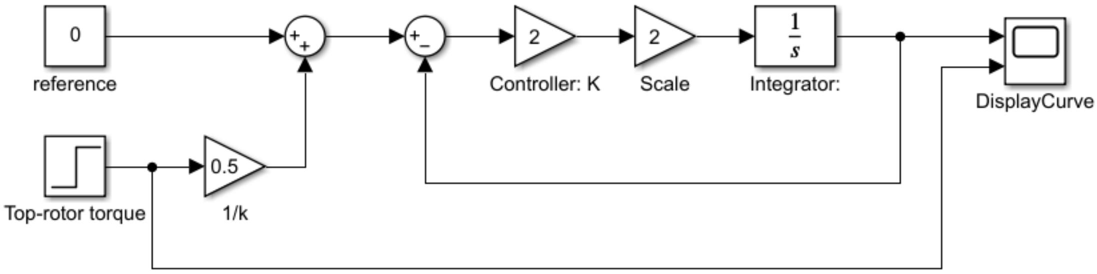
Roll No: 210447 e.g. 170001

Dept.: EE e.g. CSE

Indian Institute of Technology Kanpur
CS637 Embedded and Cyber-Physical Systems
Homework Assignment 1
Deadline: September 13, 2024

Part (a):

Plot for three different values of K, blue curve: output red curve: Top-torque input



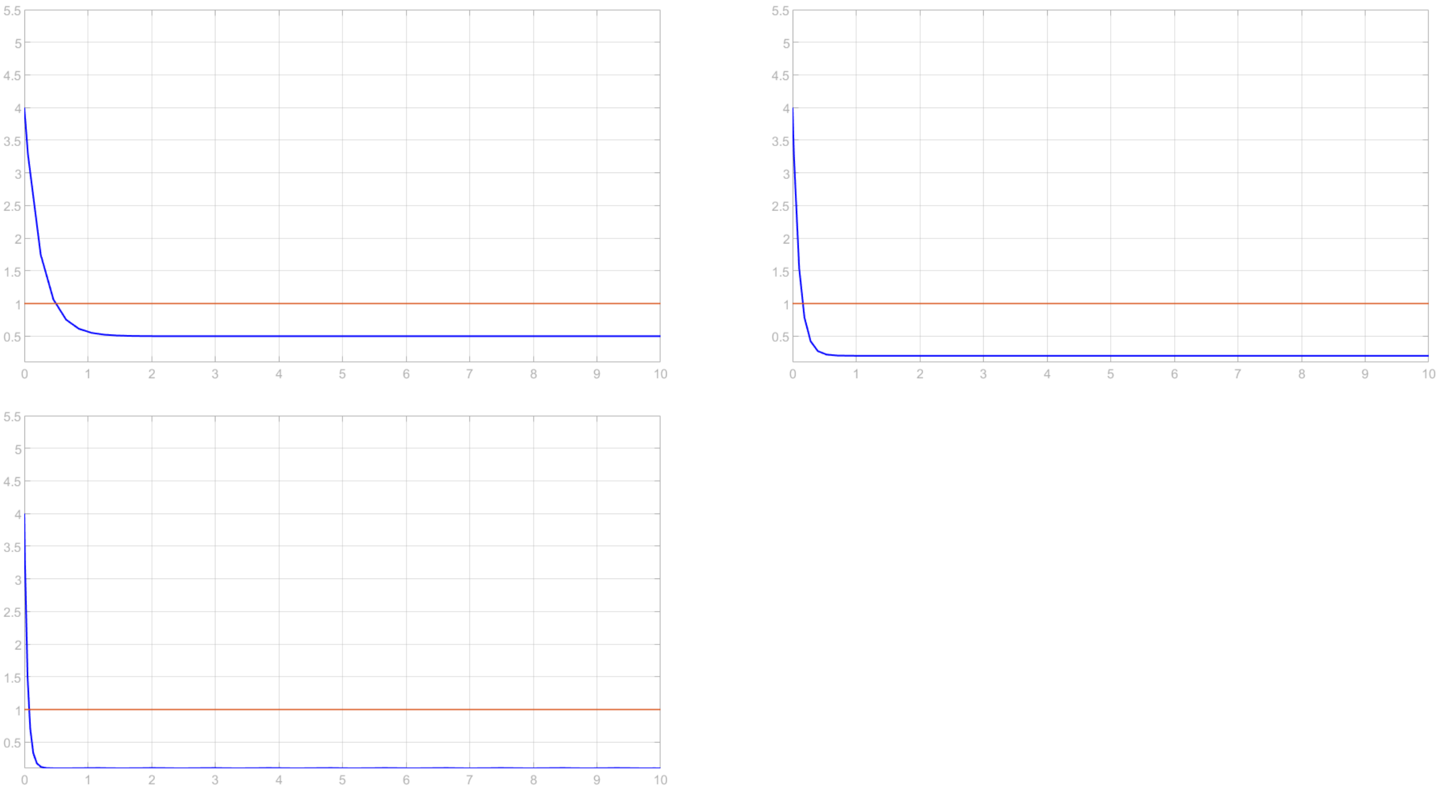


Figure 1: (i)K = 2 (ii)K = 5 (iii)K = 10

Name: Himesh Baraik

Roll No: 210447 (e.g. 170001) Dept.: EE (e.g. CSE)

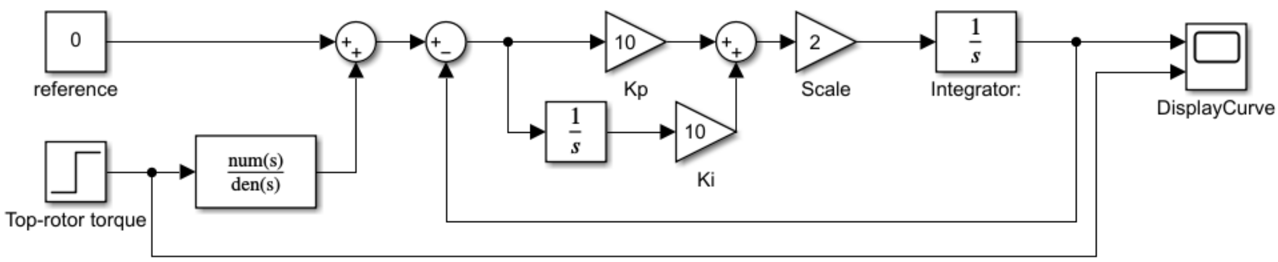
Indian Institute of Technology Kanpur
CS637 Embedded and Cyber-Physical Systems
Homework Assignment 1
Deadline: September 13, 2024

System behavior varies with K:

Reduction in Steady-State Error: The steady-state error decreases because a higher K increases the control action for a given error. This drives the system closer to the desired setpoint, reducing the error over time.

Faster Response to Steady State: A larger proportional gain also causes the system to respond more quickly to changes or disturbances. The system reaches the steady state faster because the increased control effort reduces the time taken to correct deviations from the setpoint.

Part (b):



K1 is referred as Kp and K2 as Ki
Plots for different values of Kp and Ki:

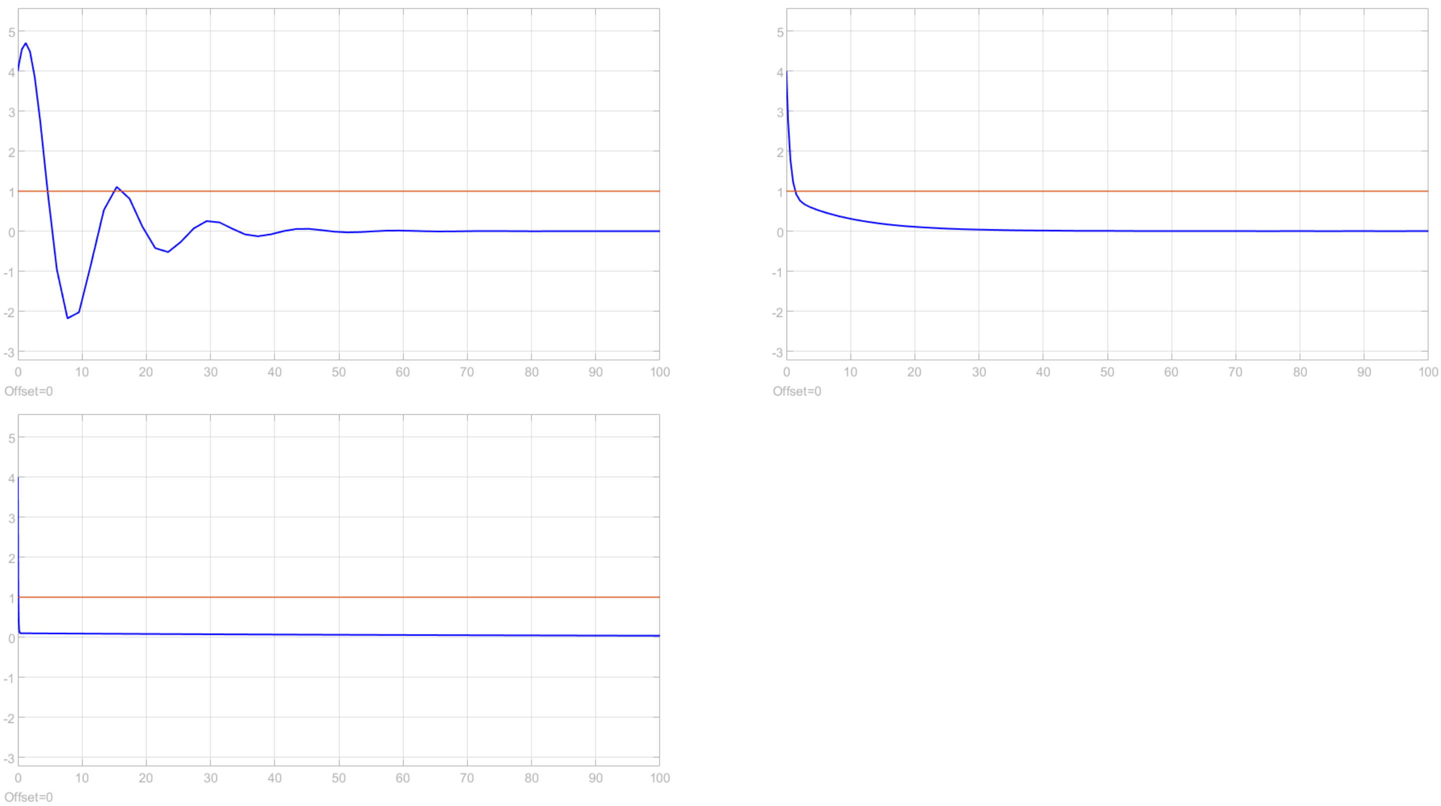


Figure 2: Kp = 0.1, (i) Ki = 0.1, (ii) ki = 1, (iii) Ki = 10

Name: Himesh Baraik

Roll No: 210447
e.g. 170001

Dept.: EE
e.g. CSE

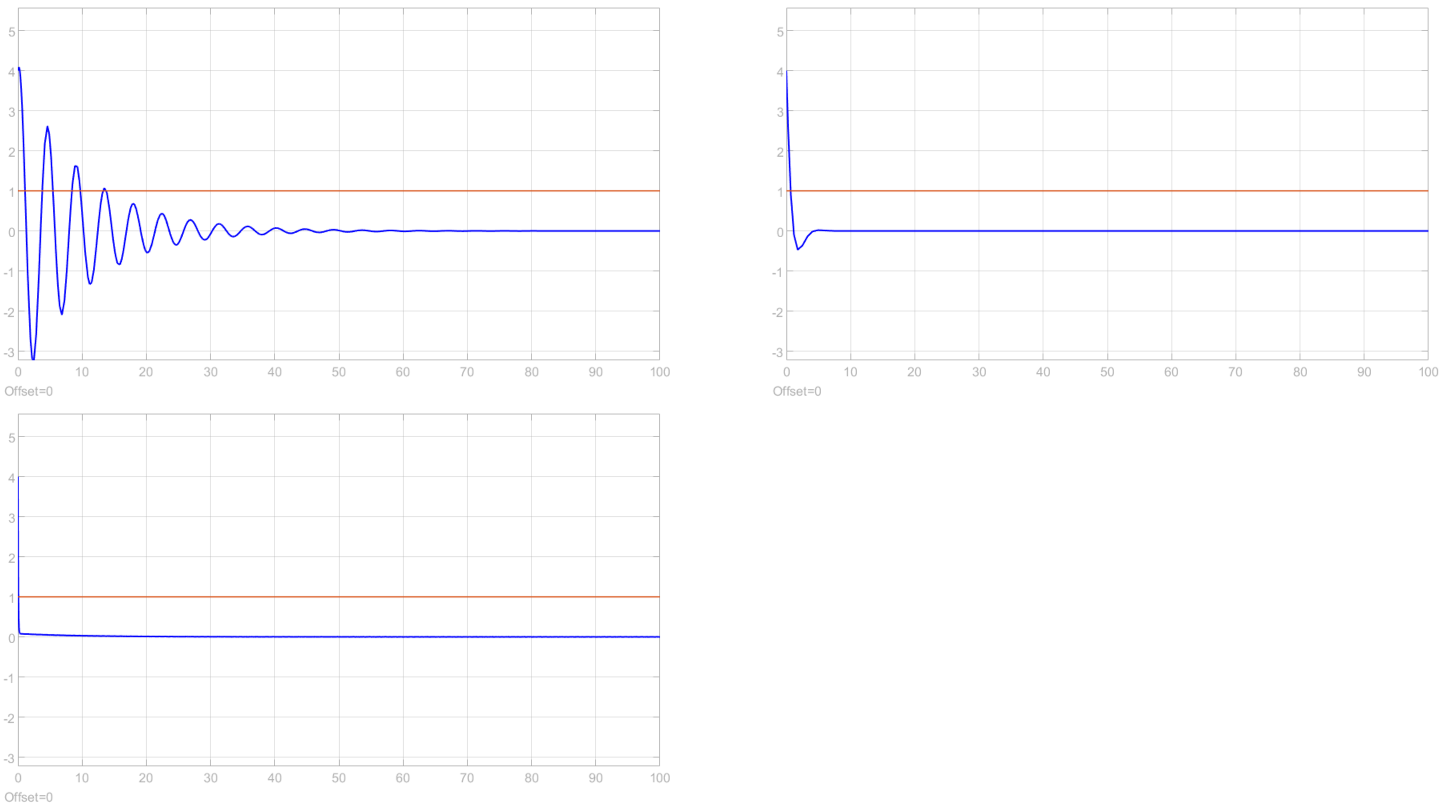


Figure 3: Kp = 1, (I) Ki = 0.1, (II) ki = 1, (III) Ki = 10

Name: Himesh Baraik

Roll No: 210447
e.g. 170001

Dept.: EE
e.g. CSE

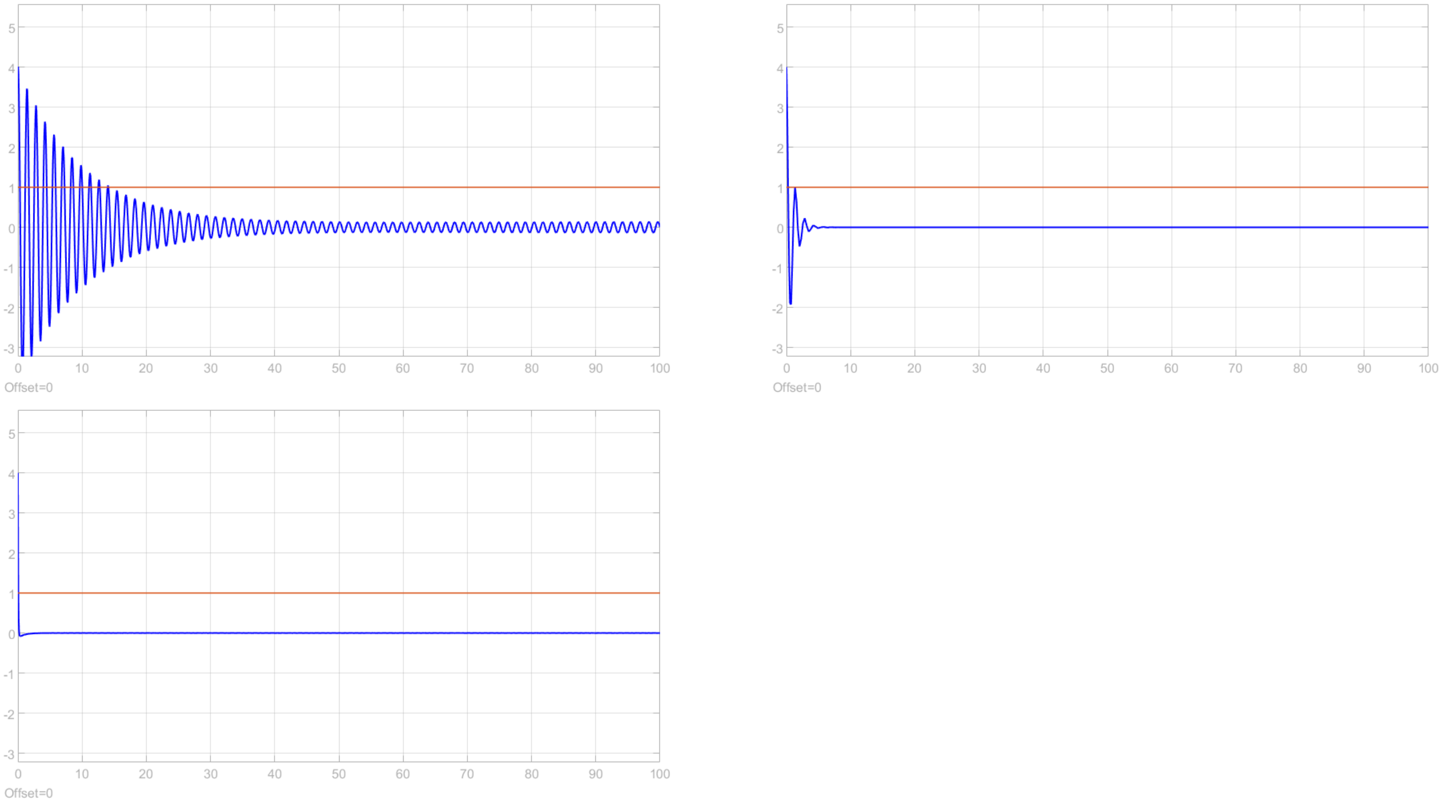


Figure 4: Kp = 10, (i) Ki = 0.1, (ii) ki = 1, (iii) Ki = 10

Obervations and comparison to controller of part (a):

Small Ki (or K2) for given Kp (or K1): The system will exhibit oscillations because the integral term takes longer to accumulate and stabilize the error.

Larger Ki (or K2) for given Kp (or K1): The system will reach steady-state faster as the integral term quickly eliminates steady-state error, but it could also lead to overshoot if Ki is too large.

The PI controller improves steady-state error performance. With a pure proportional controller, steady-state error remains unless K is large. With the PI controller, the integral action reduces steady-state error over time

Name: Himesh Baraik

Roll No: 210447 (e.g. 170001) Dept.: EE (e.g. CSE)

Indian Institute of Technology Kanpur
CS637 Embedded and Cyber-Physical Systems
Homework Assignment 1
Deadline: September 13, 2024

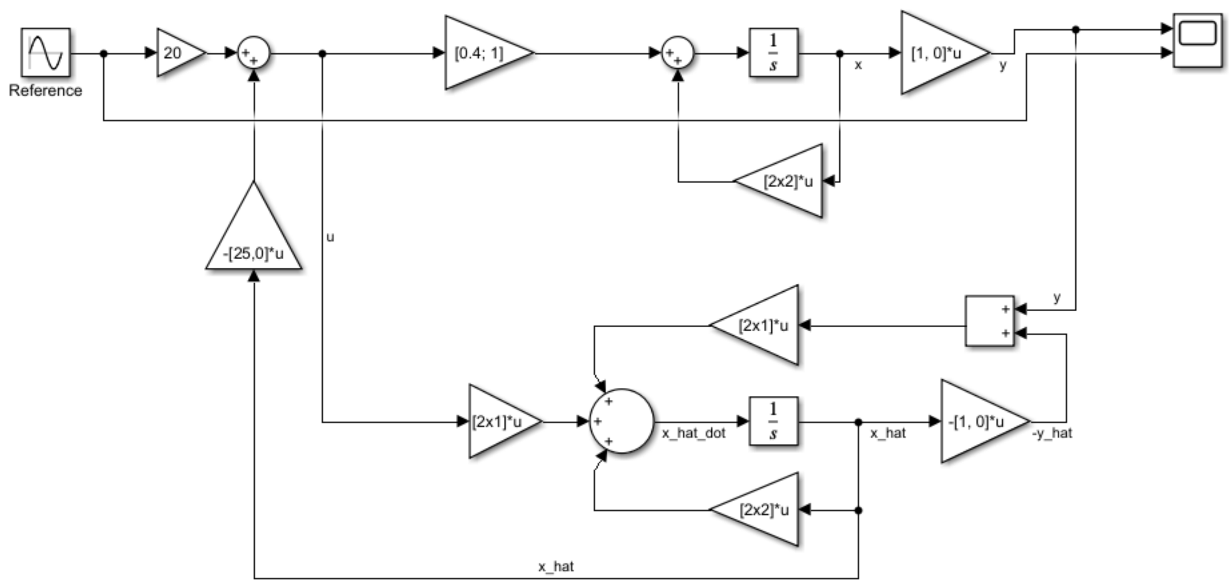
Problem 2. (10 points)

Consider the linearized model of a vehicle steering system given in Example 6.4 and its observer-based controller in Example 7.3 in [AM09]. The states of the linearized model of a vehicle steering system represent the lateral deviation of the vehicle from the x-axis and the angle between the vehicle axis and the x-axis. The output of the linearized model is only the first state. Construct a Simulink model for the vehicle steering system with its controller that includes an observer. Apply a sinusoidal signal as the reference trajectory that specifies the desired deviation of the vehicle from the x-axis with time. Plot the output (lateral deviation of the vehicle from the x-axis) with time.

[AM09] K. J. Astrom and R. M. Murray. Feedback Systems: An Introduction for Scientists and Engineers. Princeton University Press, 2009.
http://www.cds.caltech.edu/ murray/books/AM08/pdf/am08-complete_22Feb09.pdf .

Solution:

Simulation:



Parameters used:
kr = 20
A = [0 1; 0 0]
B = [0.4; 1]
C = [1, 0]
D = 0
L = [100;1000]
K = acker(A, B, [-5, -5])

Name: Himesh Baraik

Roll No: 210447
e.g. 170001

Dept.: EE
e.g. CSE

Lateral deviation of the vehicle from the x-axis) with time:

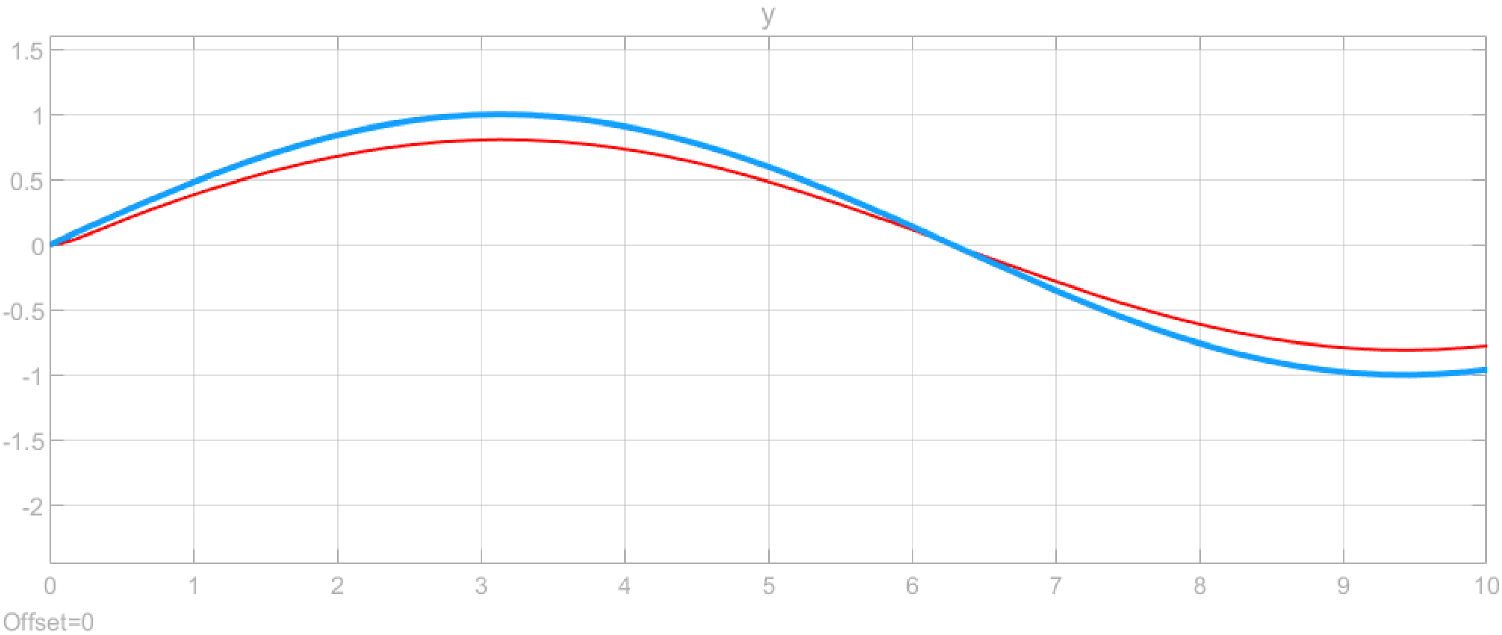


Figure 5: frequency = 0.5 rad/s

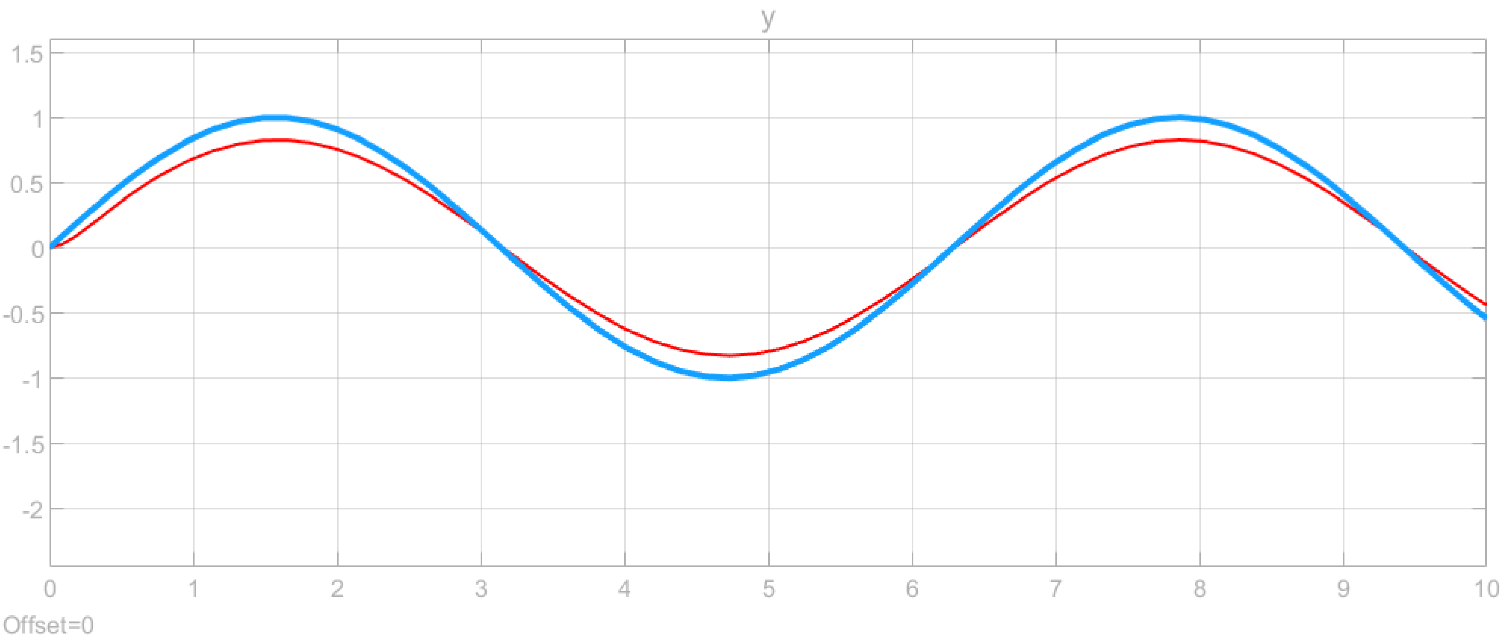


Figure 6: frequency = 1 rad/s

Name: Himesh Baraik

Roll No: 210447
e.g. 170001

Dept.: EE
e.g. CSE

Indian Institute of Technology Kanpur
CS637 Embedded and Cyber-Physical Systems
Homework Assignment 1
Deadline: September 13, 2024

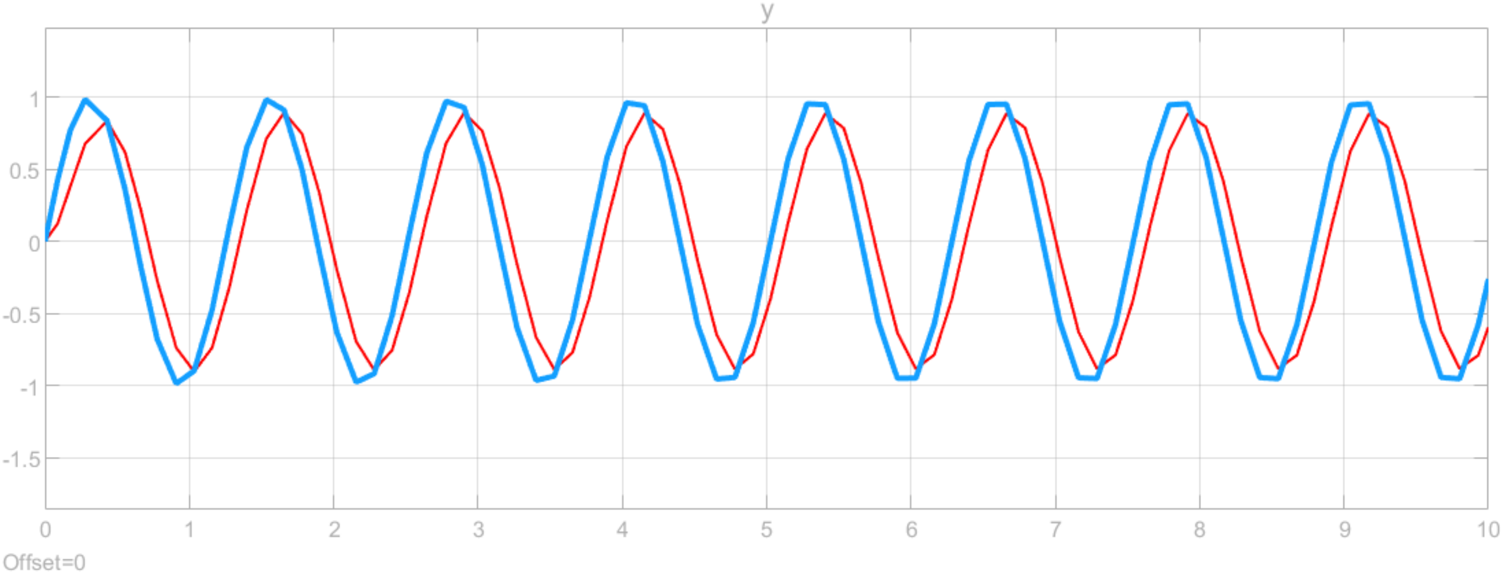


Figure 7: frequency = 5 rad/s

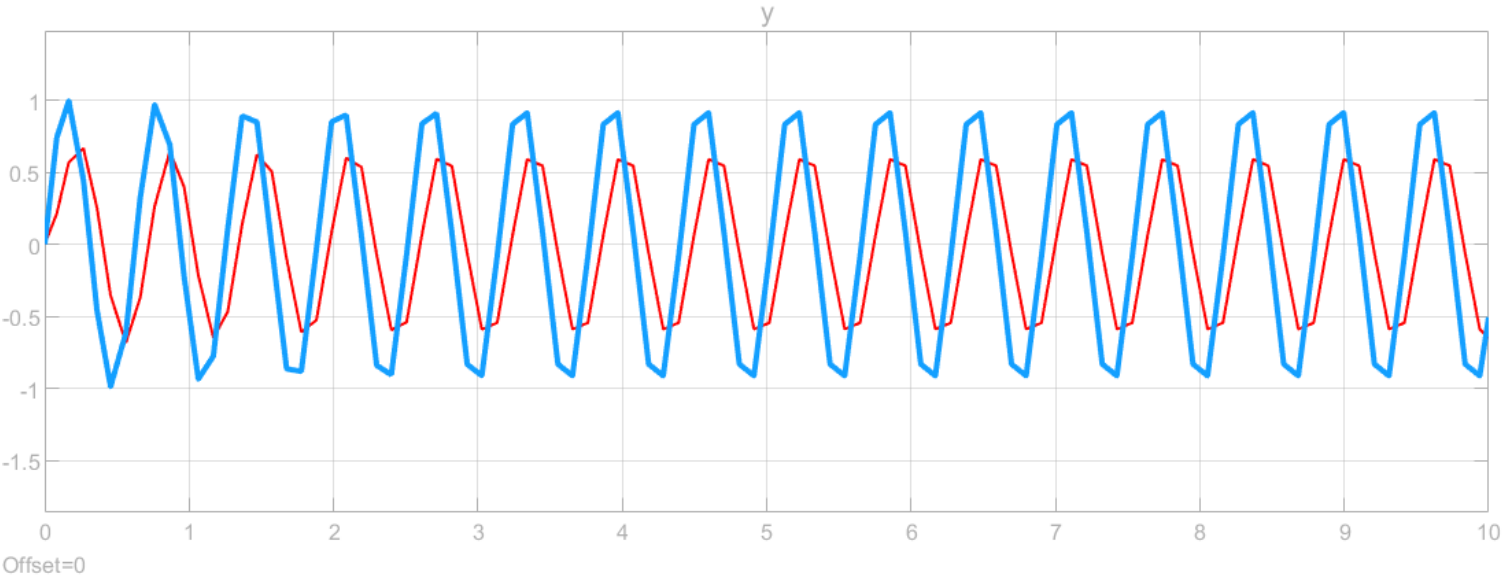


Figure 8: frequency = 10 rad/s

Name: Himesh Baraik

Roll No: 210447
e.g. 170001

Dept.: EE
e.g. CSE

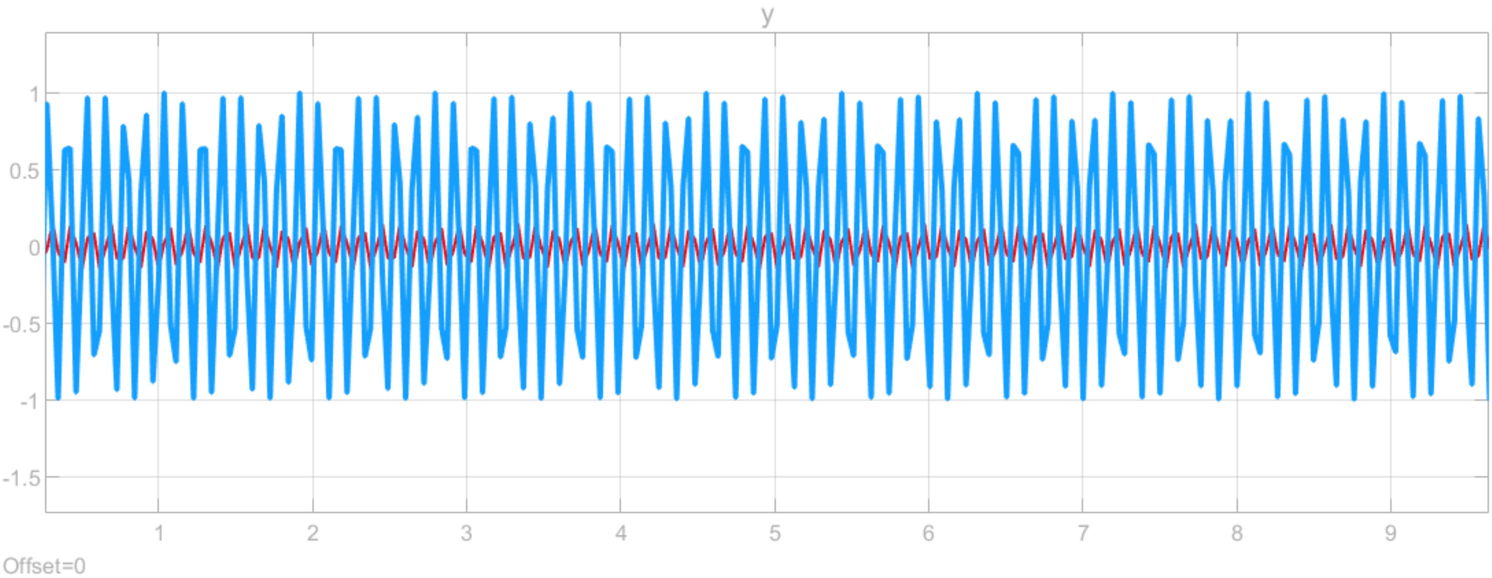


Figure 9: frequency = 50 rad/s

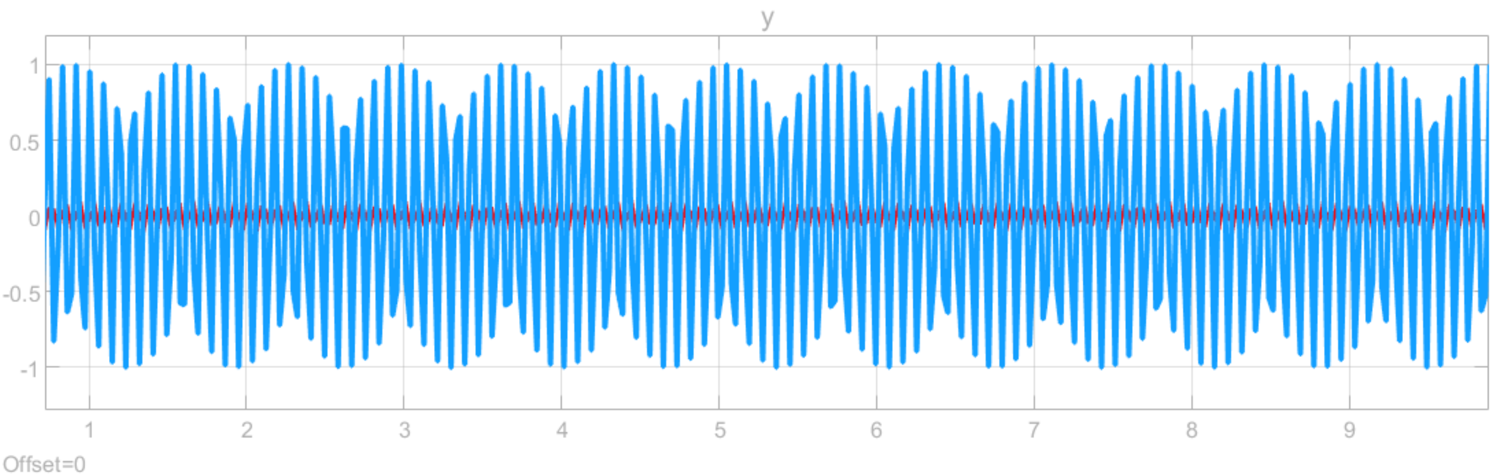


Figure 10: frequency = 70 rad/s

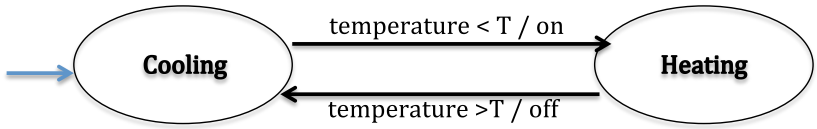
Name: Himesh Baraik

Roll No: 210447 e.g. 170001

Dept.: EE e.g. CSE

Problem 3. (10 points)

Consider the following model for a thermostat system.



The thermostat has been designed to maintain the temperature of a room at T°C. The model has two states: *cooling* and *heating*. When the system is in the *cooling* state and the temperature of the room goes below T°C, the system generates a signal to switch on a heater and moves to the *heating* state. When the temperature of the room goes over T°C, the system generates a signal to switch off the heater and moves to the *cooling* state.
(a) Represent the system as an actor that takes the current temperature as input and produces a signal to control the heater. The actor uses the set point T as a parameter.
(b) Identify a design problem in the model.
(c) Provide two different remedies to address the problem — one event triggered and one time triggered.
(d) Compare your proposed two solutions in terms of ease of implementation and guaranties on the system behavior.
(e) Provide a timed automaton model of your time-triggered solution.

Solution:

(a): Thermostat System Model

The thermostat can be modeled as an actor that takes the current temperature as an input and produces a control signal for the heater (either on or off). It operates in two states: **Cooling** and **Heating**. The parameter T is the set point temperature that the system tries to maintain.

State Machine Representation
States:

- **Cooling**: The system is in the cooling state when the temperature is above T.
- **Heating**: The system is in the heating state when the temperature is below T.

Transitions:

- From Cooling to Heating: When the temperature drops below T, the heater turns on.
- From Heating to Cooling: When the temperature rises above T, the heater turns off.

Actor Model Definition
Input: Current temperature (`Temp`)
Parameter: Set point temperature (T)
States:

- **Cooling:**
  - If `Temp` $< T$: Move to Heating, Signal = `on`
- **Heating:**
  - If `Temp` $> T$: Move to Cooling, Signal = `off`

Output: Control signal (`"on"` or `"off"`)

Name: Himesh Baraik

Roll No: 210447 (e.g. 170001)

Dept.: EE (e.g. CSE)

Indian Institute of Technology Kanpur
CS637 Embedded and Cyber-Physical Systems
Homework Assignment 1
Deadline: September 13, 2024

(b):

The design problem is **oscillatory** behavior or **chattering** around the set point T. As the temperature fluctuates slightly around T, the system may continuously switch between the "on" and "off" states, causing unnecessary wear on the heater and inefficient operation.

This can happen when the temperature fluctuates very close to T, and the system keeps turning the heater on and off frequently.

(c):

1. **Event-Triggered Solution (Hysteresis)**
   One approach to address the oscillation issue is to introduce hysteresis. This involves defining two thresholds:

   - **Lower Threshold (T_{low})**: Below this temperature, the heater turns on (heating state).
   - **Upper Threshold (T_{high})**: Above this temperature, the heater turns off (cooling state).

Event-Triggered Solution Design:

- If the temperature drops below T_{low}, the heater turns on.
- If the temperature rises above T_{high}, the heater turns off.
- No action occurs if the temperature fluctuates between T_{low} and T_{high}. This creates a buffer zone where the system doesn't react to minor fluctuations.

2. **Time-Triggered Solution (Periodic Checking)**
   Another approach is to use a time-triggered mechanism. Instead of reacting immediately to every temperature change, the system periodically checks the temperature at regular intervals (e.g., every 5 minutes).
   Time-Triggered Solution Design:

- The system reads the temperature at predefined time intervals.
- If the temperature at the time step is below T, the system turns the heater on.
- If the temperature at the time step is above T, the system turns the heater off.

This approach reduces unnecessary switching by acting only on time-sampled data and mitigates oscillations from rapid temperature changes.

Name: Himesh Baraik

Roll No: 210447
e.g. 170001

Dept.: EE
e.g. CSE

(d):

Comparison of the two solutions in terms of ease of implementation and guarantees on system behavior:

Aspect	**Event-Triggered (Hysteresis)**	**Time-Triggered (Periodic Checking)**
Ease of Implementation	Simple to implement. Requires setting two thresholds (T_{low} and T_{high}).	Slightly more complex. Requires setting up a timer and periodic temperature checking logic.
System Behavior Guarantees	Provides a clear guarantee of reduced switching by acting only when temperature is outside the hysteresis range.	Prevents excessive switching due to frequent temperature fluctuations, but there may be delayed reactions depending on the sampling interval.
Energy Efficiency	More energy efficient due to fewer frequent switches in the hysteresis range.	Could be less energy efficient if the interval is too long, leading to delayed switching on/off.
Precision	Higher precision with immediate response outside hysteresis limits.	Lower precision due to periodic checking, but still avoids small fluctuations.
Wear on Equipment	Reduces wear on equipment by preventing rapid switching between heating and cooling.	Similarly reduces wear by only switching based on periodic checks, avoiding small fluctuations.

Name: Himesh Baraik

Roll No: 210447 e.g. 170001 Dept.: EE e.g. CSE

(e) Timed Automaton Model of the Time-Triggered Solution:

The time-triggered solution can be modeled using a timed automaton, where the system periodically checks the temperature at regular intervals τ. The system switches between two states: **Cooling** (heater off) and **Heating** (heater on). A clock c is used to trigger the temperature checks.

- **States:**
  - **Cooling**: The system is in the cooling state when the heater is off.
  - **Heating**: The system is in the heating state when the heater is on.
- **Transitions:**
  - **Cooling** to **Heating**: When the clock c reaches the time interval τ and the temperature is below T, the system switches to the heating state and the clock is reset.
  - **Heating** to **Cooling**: When the clock c reaches the time interval τ and the temperature is above T, the system switches to the cooling state and the clock is reset.
- **Clocks:**
  - A clock c measures the time between temperature checks. Every τ seconds, the system reads the temperature and decides whether to switch states or remain in the current state.

The formal model of the timed automaton can be described as follows:

```
State: Cooling
  If c >=  and Temp < T -> Move to Heating, Reset c

State: Heating
  If c >=  and Temp > T -> Move to Cooling, Reset c

Clock: c
```

In this model:

- The clock c resets after every temperature check.
- The system only changes states at regular intervals, reducing the likelihood of frequent switching due to minor temperature fluctuations.

Name: Himesh Baraik

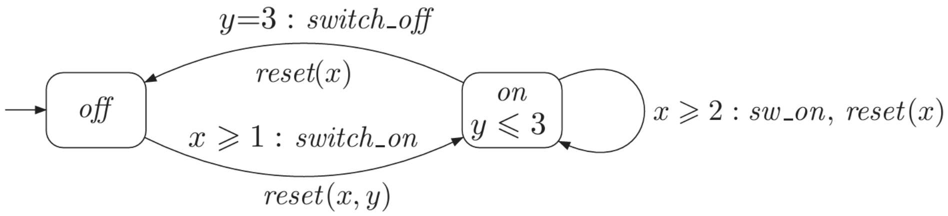
Roll No: 210447 e.g. 170001

Dept.: EE e.g. CSE

Indian Institute of Technology Kanpur
CS637 Embedded and Cyber-Physical Systems
Homework Assignment 1
Deadline: September 13, 2024

Problem 4. (10 points)

Consider the timed automaton LightSwitch illustrated below:



(a) Determine the transition system TS(LightSwitch).
(b) Check whether LightSwitch is timelock-free and non-zeno.

Solution

(a):

Transition System TS(LightSwitch)

A transition system is defined as a tuple TS $= (S, L, \Sigma, \rightarrow, I)$, where:

- S is the set of states.
- L is the set of labels.
- Σ is the set of actions.
- $\rightarrow$ is the transition relation.
- I is the set of initial states.

For the LightSwitch automaton, we have:

- States (S): { `off`, `on` }
- Labels (L): { $x \geq 1$, $x \geq 2$, $y = 3$, $y \leq 3$ }
- Actions (Σ): { `switch_on`, `switch_off`, `reset`(x), `reset`(x, y) }
- Transition Relation ($\rightarrow$): Defined by transitions in the diagram. For example:
  - $(\texttt{off}, x \geq 1, \texttt{switch\_on}, \texttt{on})$
  - $(\texttt{on}, y \leq 3, \texttt{switch\_off}, \texttt{off})$
  - Additional transitions would be similarly defined based on the automaton's diagram.
- Initial State (I):{ `off` }

Name: Himesh Baraik

Roll No: 210447 (e.g. 170001)   Dept.: EE (e.g. CSE)

(b):

Timelock-Freedom and Non-Zenoness

Timelock-Freedom

A timed automaton is **timelock-free** if, for every reachable state, and for every possible delay, there exists a transition that can be taken. In other words, the automaton cannot get stuck due to the passage of time.

- In the `off` state, transitions are enabled if $x \geq 1$ or $y = 3$. Hence, there is no state where the automaton gets stuck due to time passage.
- In the `on` state, transitions are enabled if $x \geq 2$ or $y \leq 3$. Thus, every state has transitions that are not blocked by the passage of time.

Thus, LightSwitch is **timelock-free**.

Non-Zenoness

A timed automaton is **non-zeno** if it cannot perform an infinite number of actions in a finite amount of time. This is ensured if there are no transitions that can be taken infinitely many times within a finite time.

- The automaton's clock reset actions are bounded by conditions:
  - `reset`(x) occurs under conditions $x \geq 1$ or $x \geq 2$.
  - `reset`(x, y) occurs under conditions $y = 3$ or $y \leq 3$.
- These conditions ensure that the automaton cannot continuously reset clocks to perform an infinite number of transitions within finite time.

Thus, LightSwitch is **non-zeno**.

Conclusion

The timed automaton LightSwitch is both **timelock-free** and **non-zeno**. This means that it is well-behaved with respect to its timing constraints, progressing without getting stuck or exhibiting infinite transitions within finite time.

Name: Himesh Baraik

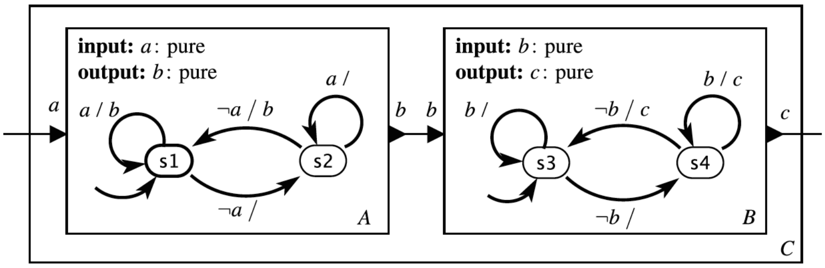
Roll No: 210447 (e.g. 170001)

Dept.: EE (e.g. CSE)

Indian Institute of Technology Kanpur
CS637 Embedded and Cyber-Physical Systems
Homework Assignment 1
Deadline: September 13, 2024

Problem 5. (10 points)

Consider the following synchronous composition of two state machines A and B.



Construct a single state machine C representing the composition. Which states of the composition are unreachable?

Solution:

States of State Machine C:

- (s1, s3)
- (s1, s4)
- (s2, s3)
- (s2, s4)

Transitions of State Machine C:

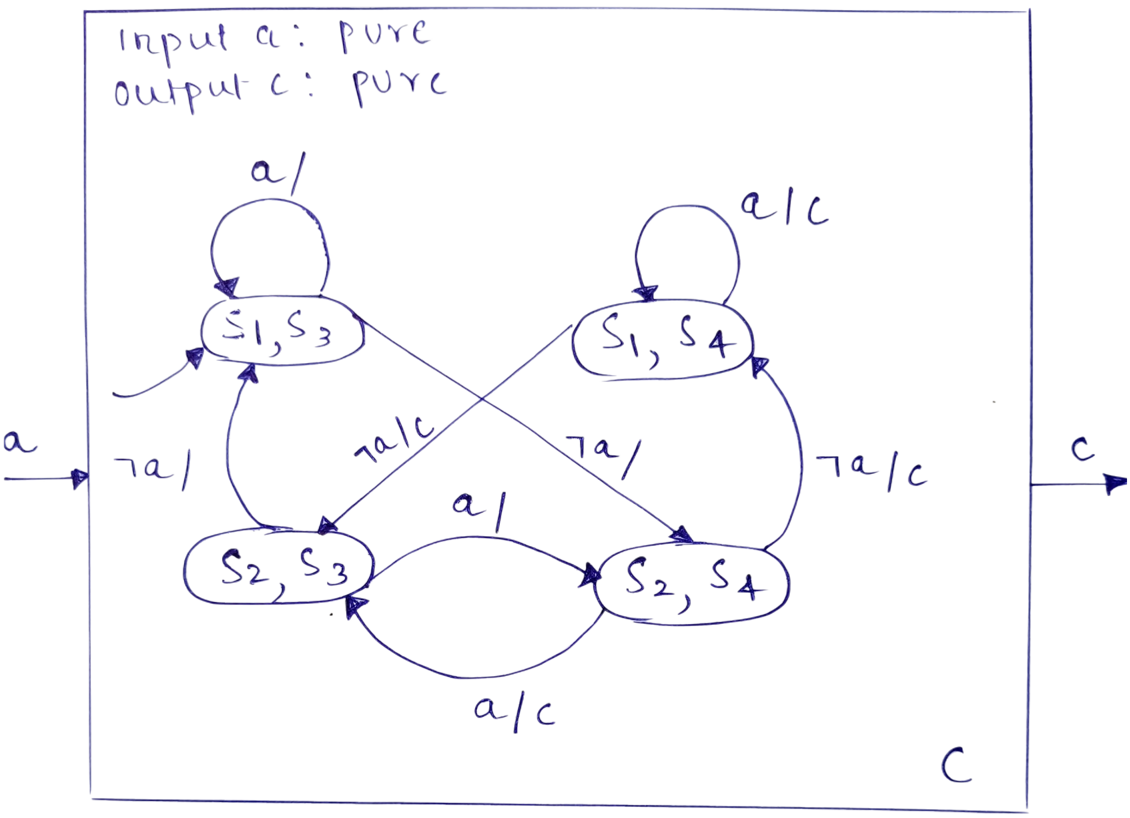
- (s1, s3) $\rightarrow$ (s1, s3): a/
- (s1, s4) $\rightarrow$ (s1, s4): a/c
- (s1, s4) $\rightarrow$ (s2, s3): -a/c
- (s1, s3) $\rightarrow$ (s2, s4): -a/
- (s2, s3) $\rightarrow$ (s1, s3): -a/
- (s2, s4) $\rightarrow$ (s1, s4): -a/c
- (s2, s4) $\rightarrow$ (s2, s3): a/c
- (s2, s3) $\rightarrow$ (s2, s4): a/

Name: Himesh Baraik

Roll No: 210447 (e.g. 170001)

Dept.: EE (e.g. CSE)

Indian Institute of Technology Kanpur
CS637 Embedded and Cyber-Physical Systems
Homework Assignment 1
Deadline: September 13, 2024



Identifying Unreachable States:
An unreachable state is a state that cannot be reached from the initial state under any sequence of inputs. In this case, the initial state is assumed to be (s1, s3).
By analyzing the transitions in C, we can observe that all states are reachable from (s1, s3). Therefore, there are no unreachable states in the composed state machine C.

Conclusion:
The synchronous composition of state machines A and B results in a composed state machine C with four states. All states in C are reachable from the initial state, making none of them unreachable.