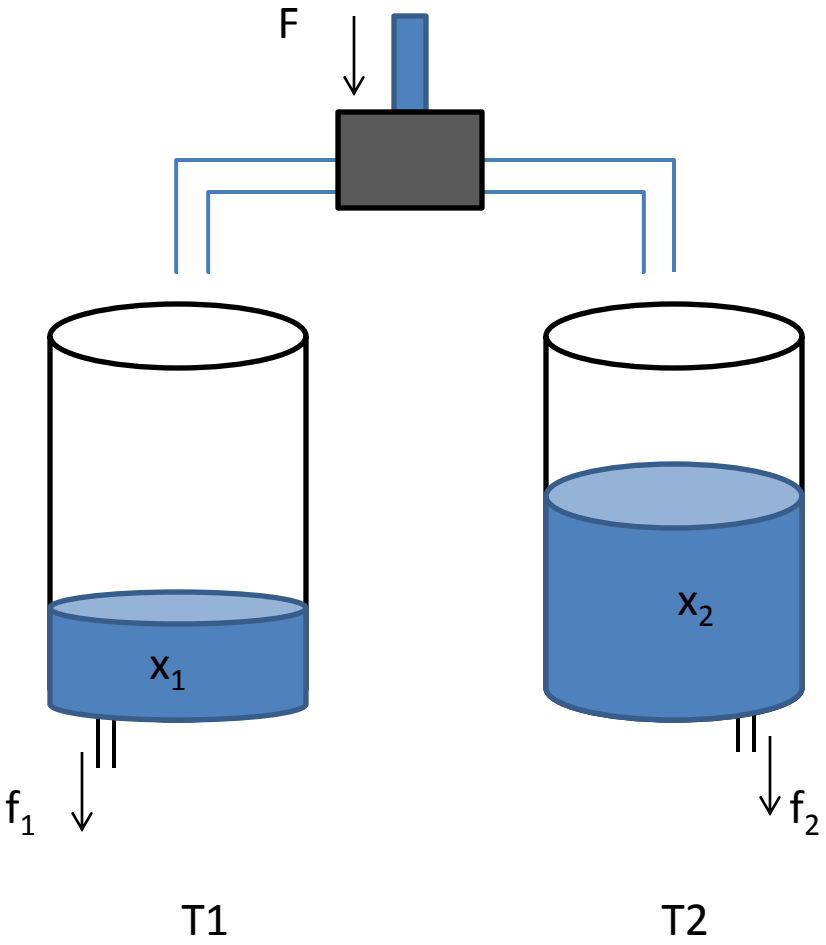
Embedded and Cyber-Physical Systems

- Modeling of continuous dynamics-

Reference book: Edward A. Lee and Pravin Varaiya, Structure and Interpretation of Signals and Systems, Second Edition, LeeVaraiya.org, ISBN 978-0-578-07719-2, 2011.

A two-tank example

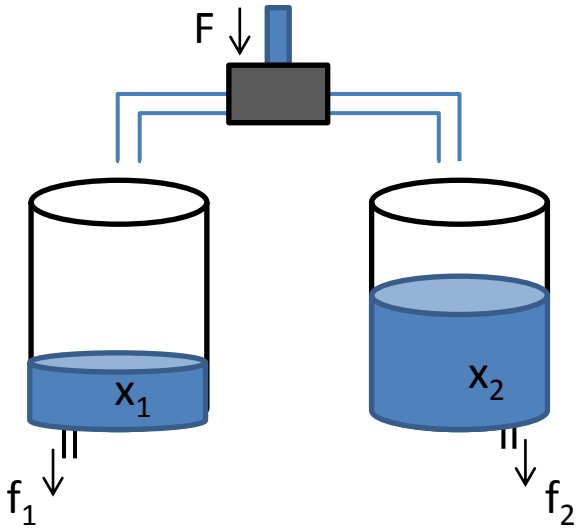


The problem

- Setting:
  - F(t)= c (constant)
  - $f_1(0) = f_2(0) = 0$
  - $x_1(0) > 0, x_2(0) > 0$
  - $f_1(t) = c_1$, $f_2(t) = c_2$ for $t > 0$
  - $c_1 < c$ and $c_2 < c$
  - F can be switched either to T1 or to T2
- Requirement: find a suitable switching to keep both levels above zero.

The **Cyber** Part

- Possible solution: a discrete controller
  - if (when) x_i == 0, switch to T_i (i=1,2)
- Advantage: Minimally interventive
- Is this a viable controller?



The **Physical** Part

- Principle of mass conservation (assume isolated system, no work)

  Input mass = stored mass + output mass

  $F \cdot t = M_1(t) + M_2(t) + (f_1 + f_2) \cdot t$

  $$F = \frac{dm}{dt}$$

  $M_1(t) + M_2(t) = (F - f_1 - f_2) \cdot t$

  What if $(F - f_1 - f_2) < 0$?

A mathematical model

$$(1)\begin{cases} \dot{x}_1 = F - f_1 \\ \dot{x}_2 = -f_2 \end{cases}$$

$$(2)\begin{cases} \dot{x}_1 = -f_1 \\ \dot{x}_2 = F - f_2 \end{cases}$$

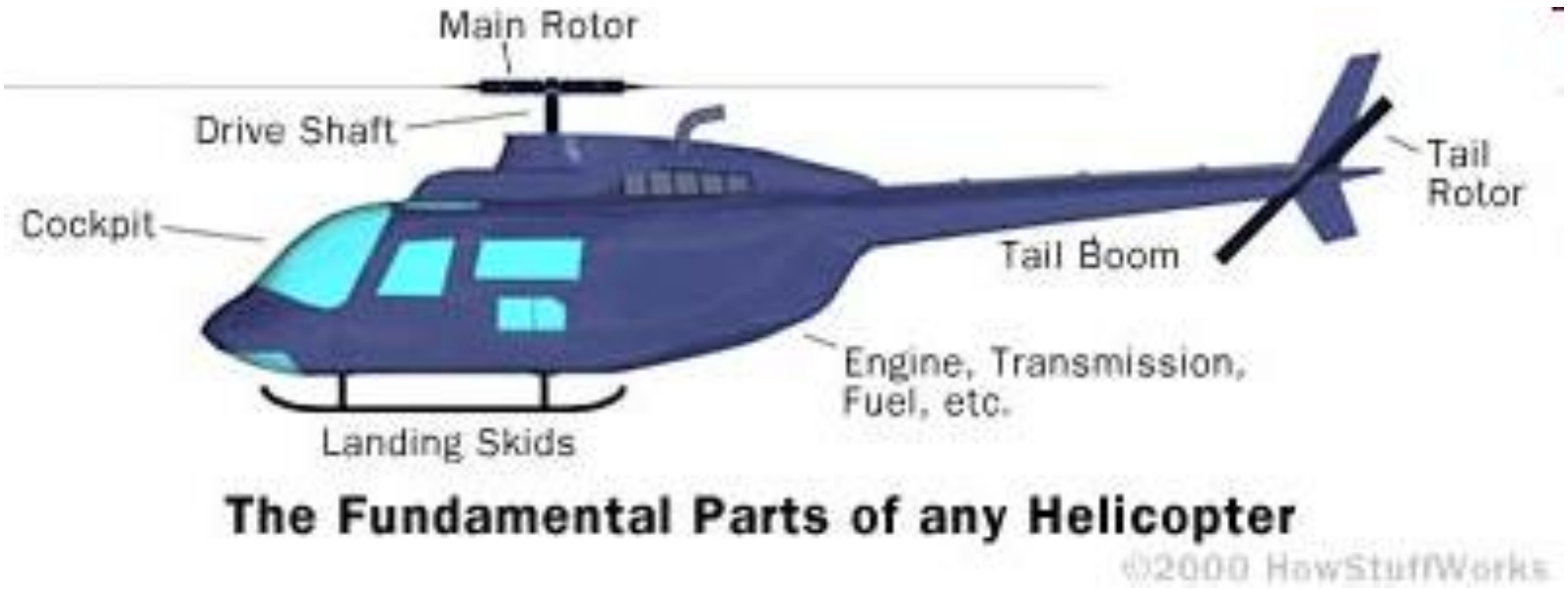
Solution

$$\begin{cases} x_1(t) = x_1^0 + (F - f_1)\cdot \Delta t_1 + (-f_1)\cdot \Delta t_2 \\ x_2(t) = x_2^0 + (-f_2)\cdot \Delta t_1 + (F - f_2)\cdot \Delta t_2 \end{cases}$$

Does a viable controller exist?
Check this:

$$\begin{cases} (F - f_1)\cdot \Delta t_1 + (-f_1)\cdot \Delta t_2 \geq 0 \\ (-f_2)\cdot \Delta t_1 + (F - f_2)\cdot \Delta t_2 \geq 0 \end{cases}$$

Another Example: Modeling Helicopter Dynamics



Feedback Control Problem

A helicopter without a tail rotor, like the one below, will spin uncontrollably. Why?

Control system problem: Apply torque using the tail rotor.

Physics teaser: helicopter spin

- Watch the two videos, with helicopters that are similar in construction.

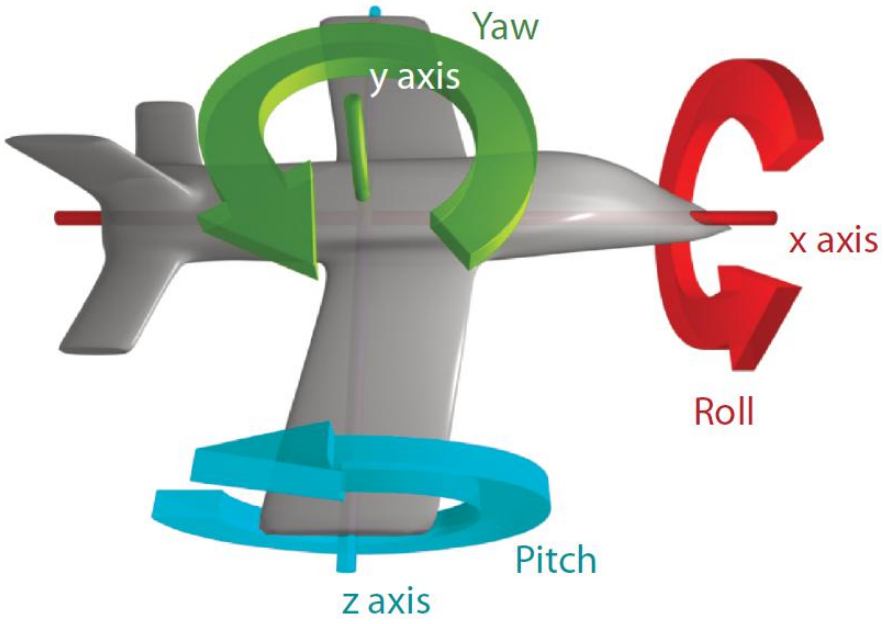
http://www.youtube.com/watch?v=MhEXXgiIVuY

http://www.youtube.com/watch?v=ijErOpjx_Ws

- Why do they behave differently after loosing the tail rotor control?

Modeling Physical Motion

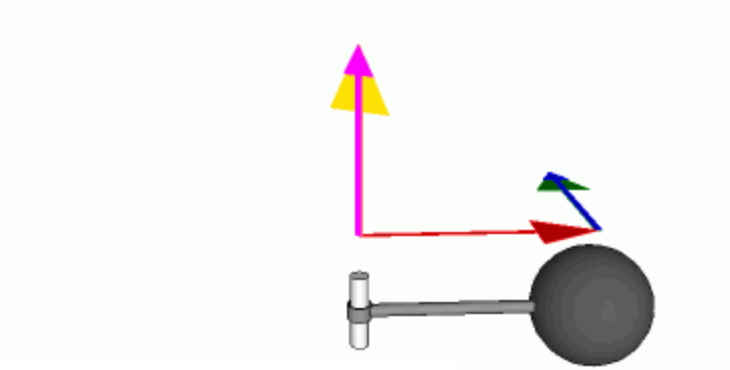
- Six degrees of freedom:
  - Position: x, y, z
  - Orientation: pitch, yaw, roll



Torque

For a point mass rotating around a fixed axis:

- radius of the arm: $r \in \mathbb{R}$
- force orthogonal to arm: $f \in \mathbb{R}$



$$T_y(t) = r\, f(t)$$

- Just as force is a push or a pull, a torque is a twist.
- Units: Newton-meters/radian, Joules/radian

Rotational Version of Newton's Second Law

$$\mathbf{T}(t) = \frac{d}{dt}\left(I(t)\dot{\theta}(t)\right),$$

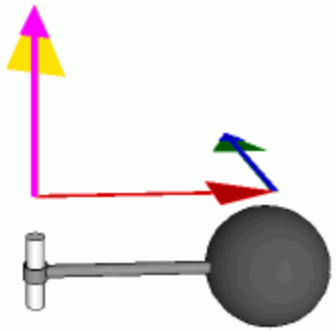
where $I(t)$ is a 3×3 matrix called the moment of inertia tensor.

$$\begin{bmatrix} T_x(t) \\ T_y(t) \\ T_z(t) \end{bmatrix} = \frac{d}{dt}\left(\begin{bmatrix} I_{xx}(t) & I_{xy}(t) & I_{xz}(t) \\ I_{yx}(t) & I_{yy}(t) & I_{yz}(t) \\ I_{zx}(t) & I_{zy}(t) & I_{zz}(t) \end{bmatrix}\begin{bmatrix} \dot{\theta}_x(t) \\ \dot{\theta}_y(t) \\ \dot{\theta}_z(t) \end{bmatrix}\right)$$

Here, for example, $T_y(t)$ is the net torque around the y axis (which would cause changes in yaw), $I_{yx}(t)$ is the inertia that determines how acceleration around the x axis is related to torque around the y axis.

Slide from Lee & Seshia

Simple Example



Yaw dynamics:

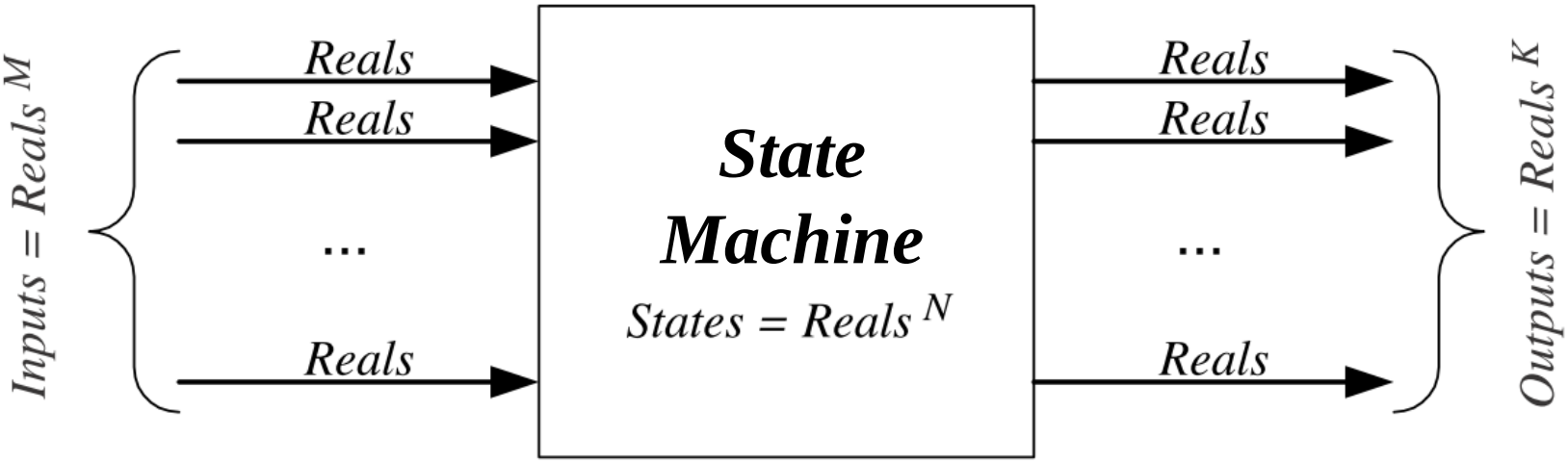
$$T_y(t) = I_{yy}\ddot{\theta}_y(t)$$

To account for initial angular velocity, write as

$$\dot{\theta}_y(t) = \dot{\theta}_y(0) + \frac{1}{I_{yy}} \int_0^t T_y(\tau) d\tau.$$

Models

Discrete model of systems



$$StateMachine = (States, Inputs, Outputs, update, initialState)$$

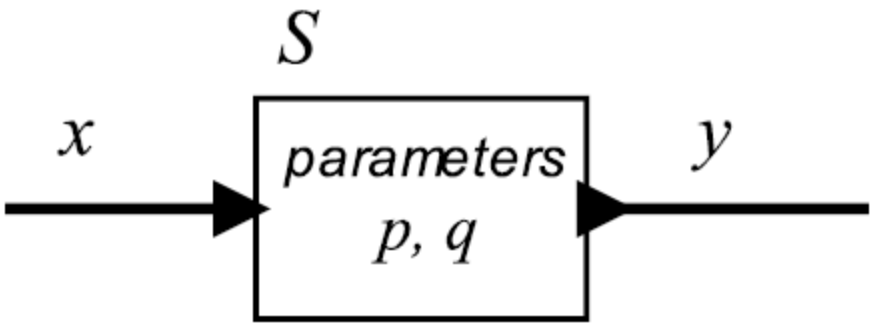
$s(n)$ $x(n)$ $y(n)$ $s(0)$

$$s(0) = initialState,$$

$$\forall n \geq 0,\ (s(n+1), y(n)) = update(s(n), x(n))$$

Actor Model of Systems

- A *system* is a function that accepts an input *signal* and yields an output signal.



- The domain and range of the system function are sets of signals, which themselves are functions.

$$x: \mathbb{R} \rightarrow \mathbb{R}, \quad y: \mathbb{R} \rightarrow \mathbb{R}$$

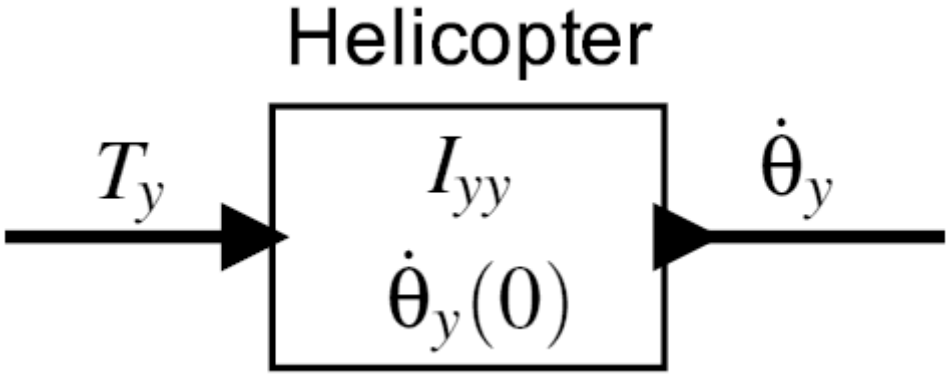
$$S: X \rightarrow Y$$

$$X = Y = (\mathbb{R} \rightarrow \mathbb{R})$$

- Parameters may affect the definition of the function *S*.

Actor model of the helicopter

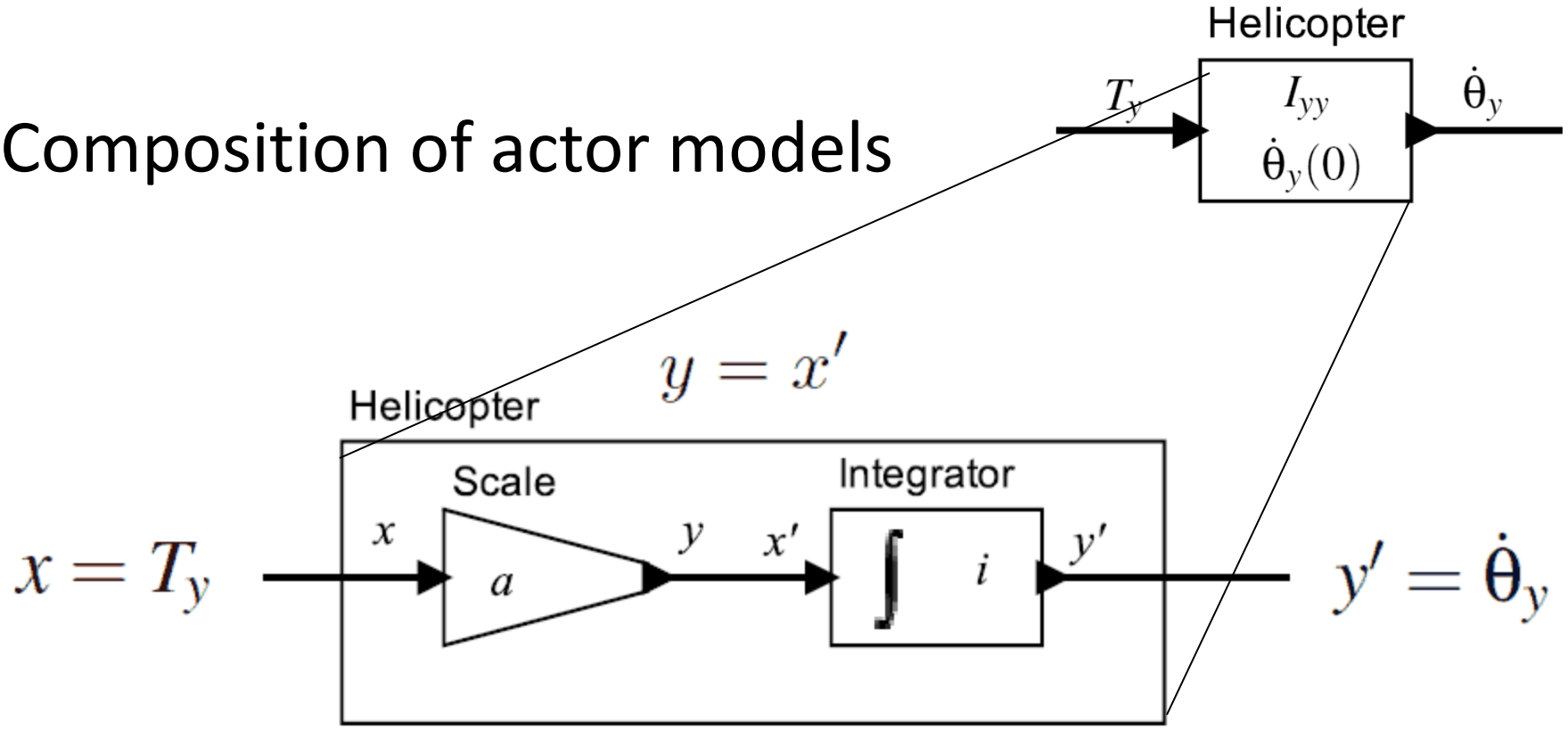
- Input is the net torque of the tail rotor. Output is the angular velocity around the *y* axis.



Parameters of the model are shown in the box. The input and output relation is given by the equation to the right.

$$\dot{\theta}_y(t) = \dot{\theta}_y(0) + \frac{1}{I_{yy}} \int_0^t T_y(\tau) d\tau$$

Composition of actor models



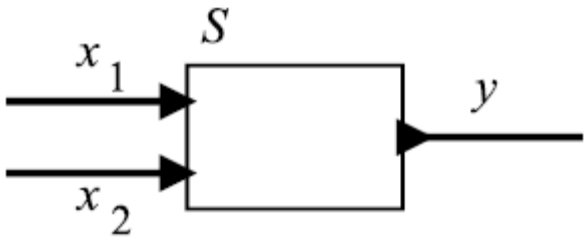
$$\forall\, t \in \mathbb{R}, \quad y(t) = ax(t) \qquad y'(t) = i + \int_0^t x'(\tau)d\tau$$

$$y = ax$$

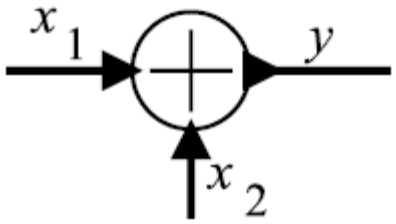
$$i = \dot{\theta}_y(0)$$

$$a = 1/I_{yy}$$

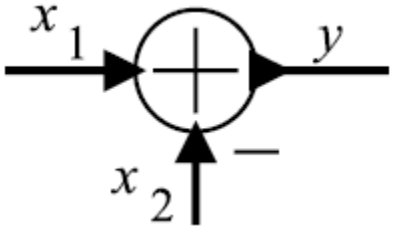
Actor models with multiple inputs



$$S: (\mathbb{R} \to \mathbb{R})^2 \to (\mathbb{R} \to \mathbb{R})$$

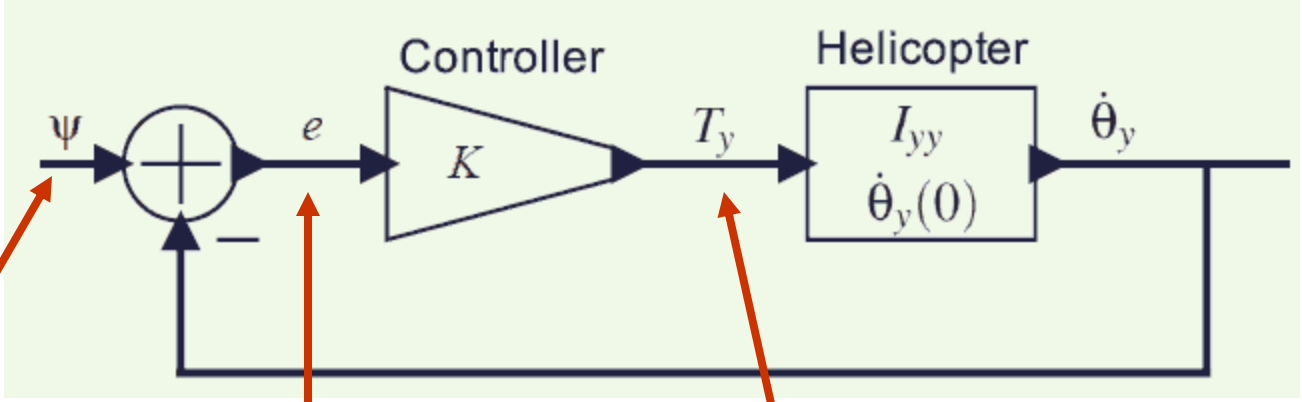


$$\forall\, t \in \mathbb{R}, \quad y(t) = x_1(t) + x_2(t)$$



$$(S(x_1, x_2))(t) = y(t) = x_1(t) - x_2(t)$$

Proportional controller



desired angular velocity

error signal

net torque

$$e(t) = \psi(t) - \dot{\theta}_y(t)$$

$$T_y(t) = Ke(t)$$

$$\dot{\theta}_y(t) = \dot{\theta}_y(0) + \frac{1}{I_{yy}} \int_0^t T_y(\tau) d\tau$$

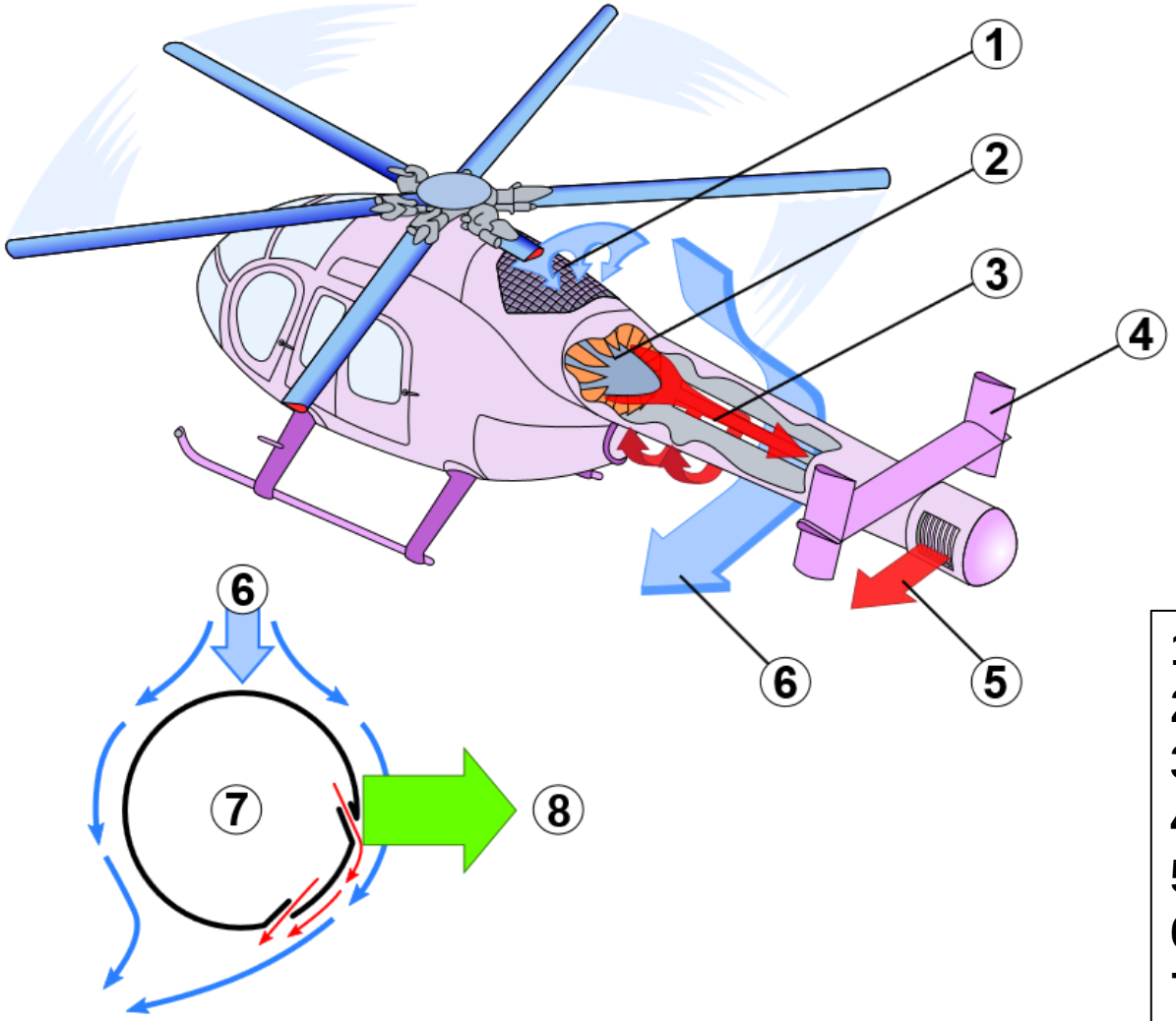
$$= \dot{\theta}_y(0) + \frac{K}{I_{yy}} \int_0^t (\psi(\tau) - \dot{\theta}_y(\tau)) d\tau$$

Note that the angular velocity appears on both sides, so this equation is not trivial to solve.

Helicopter question

- What other effect has the tail rotor torque?
- Think about how to counteract that
  - New control strategy, new actuator on the helicopter

NOTAR helicopter



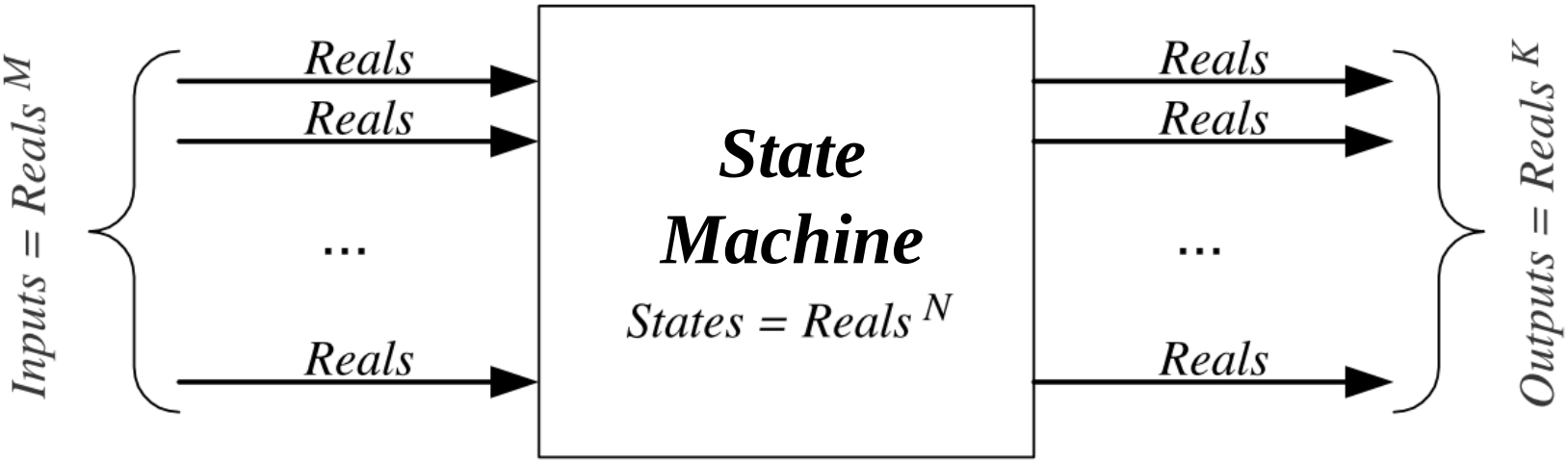
1 Air intake
2 Variable pitch fan
3 Tail boom with Coandă Slots
4 Vertical stabilizers
5 Direct jet thruster
6 Downwash
7 Tailboom cross-section
8 Anti-torque lift

"NOTAR System" picture by Voytek S

Embedded and Cyber-Physical Systems

- Modeling discrete behavior -

Discrete model of systems



$$StateMachine = (States, Inputs, Outputs, update, initialState)$$

$s(n)$ $x(n)$ $y(n)$ $s(0)$

$$s(0) = initialState,$$

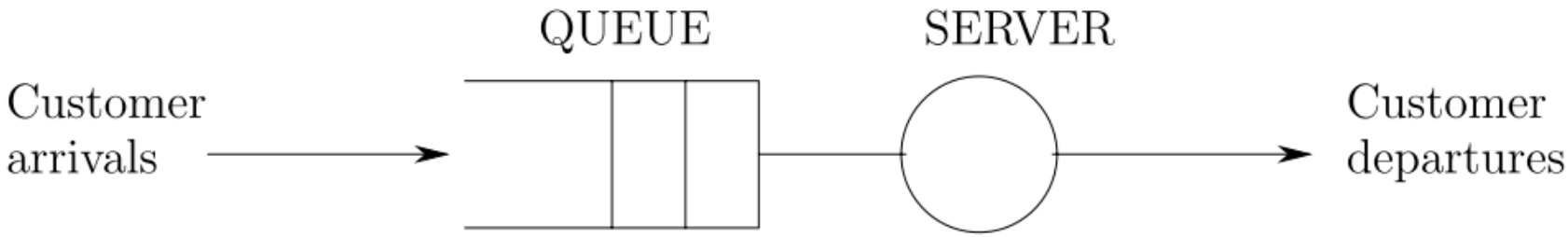
$$\forall n \geq 0,\ (s(n+1), y(n)) = update(s(n), x(n))$$

Discrete Event Systems

- *States* is a discrete (finite or countable) set
- *Inputs* and *outputs* are event sequences
- The *update* function is a transition between discrete states

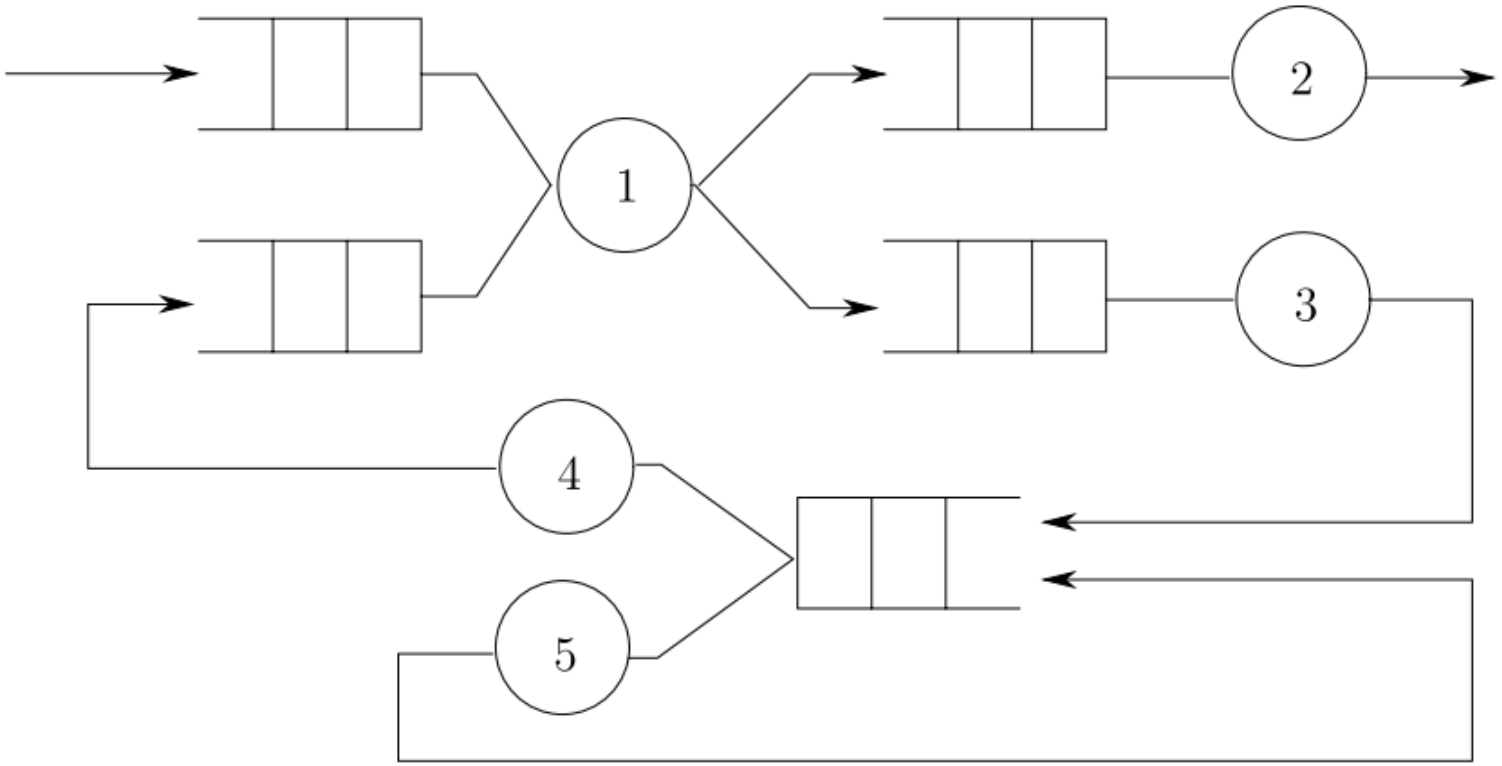
- The system reacts to discrete events, not to the passage of time

Example: Queueing systems



- Elements:
  - Capacity
  - Queueing policy
  - Arrival event
  - Departure event
- What are the states?

Example: Queueing networks

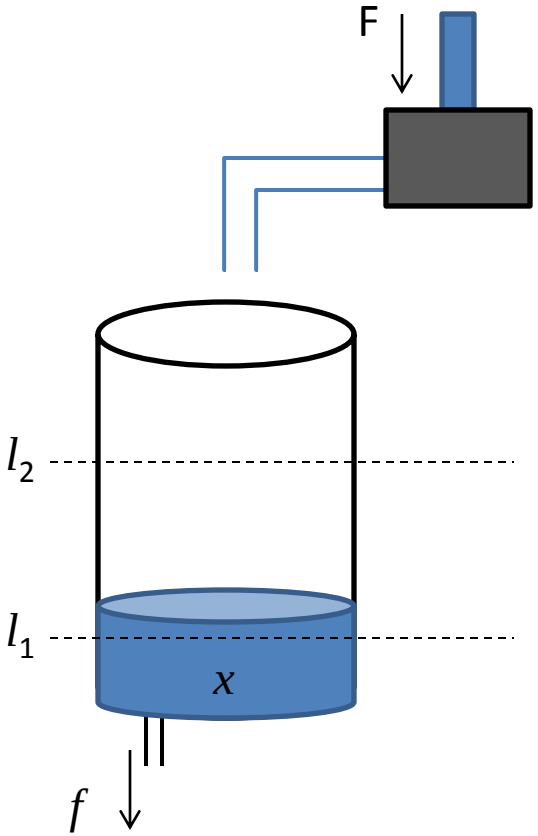


Other examples

- Computer systems (jobs competing for processors and peripherals)
- Manufacturing systems (production parts competing for machines)
- Traffic systems (vehicles competing for space)
- Database systems (maintaining consistency under concurrent transactions)
- *All the above are human artifacts!*

DES as abstractions of continuous systems

- Another tank example
  - A tank with variable outflow
  - A 2-speed pump
  - Keep level between l_1 and l_2



- What are the states?
- What are the events?
- Do we need to represent time?