Cyber-Physical Systems

Modeling Physical Dynamics

IECE 553/453– Fall 2021

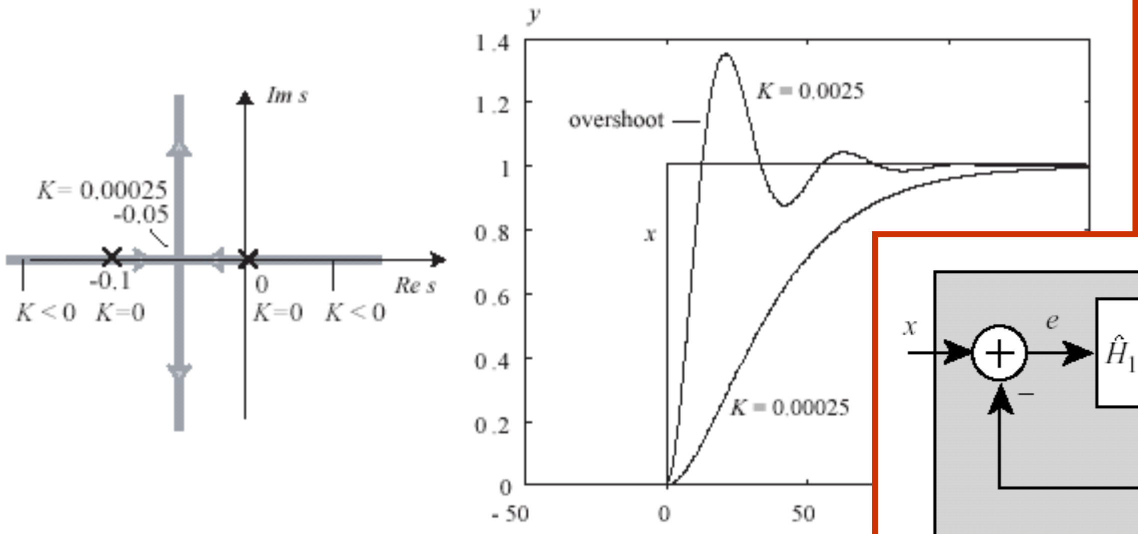
Prof. Dola Saha

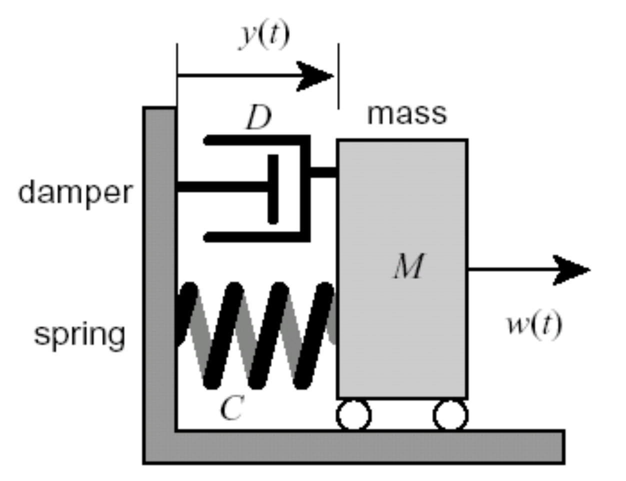
Modeling Techniques

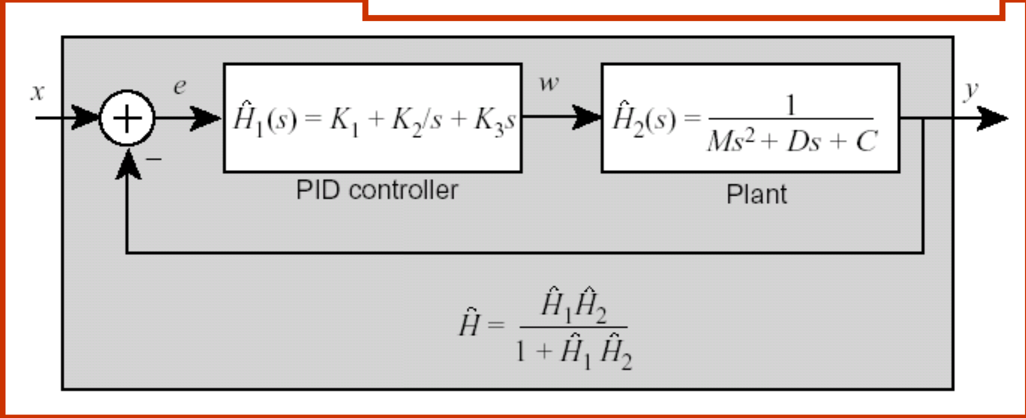
- Models that are abstractions of **system dynamics** (how system behavior changes over time)
  - Modeling physical phenomena – differential equations
  - Feedback control systems – time-domain modeling
  - Modeling modal behavior – FSMs, hybrid automata, …
  - Modeling sensors and actuators –calibration, noise, …
  - Hardware and software – concurrency, timing, power, …
  - Networks – latencies, error rates, packet losses, …

Modeling of Continuous Dynamics

- Ordinary differential equations, Laplace transforms, feedback control models, …

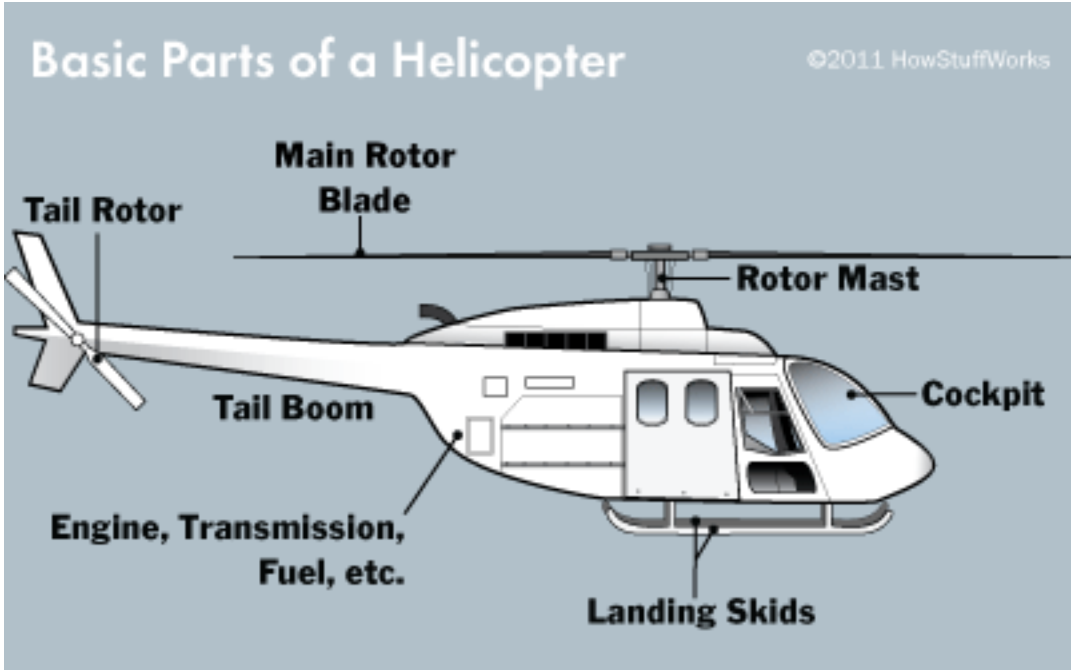






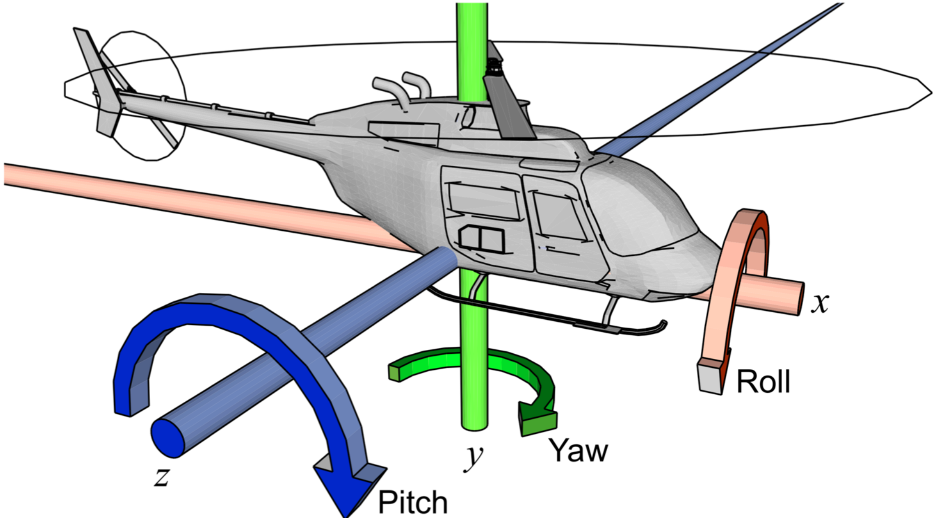
Example CPS System

- Helicopter Dynamics



Modeling Physical Motion

- Six Degrees of Freedom
  - Position: x, y, z
  - Orientation: roll (θ_x), yaw (θ_y), pitch (θ_z)



Notation

Position is given by three functions:

$$x: \mathbb{R} \rightarrow \mathbb{R}$$
$$y: \mathbb{R} \rightarrow \mathbb{R}$$
$$z: \mathbb{R} \rightarrow \mathbb{R}$$

where the domain $\mathbb{R}$ represents time and the co-domain (range) $\mathbb{R}$ represents position along the axis. Collecting into a vector:

$$\mathbf{x}: \mathbb{R} \rightarrow \mathbb{R}^3$$

Position at time $t \in \mathbb{R}$ is $\mathbf{x}(t) \in \mathbb{R}^3$.

Orientation can be represented in the same form

- Functions of this form are known as continuous-time signals

Notation

Velocity

$$\dot{\mathbf{x}}\colon \mathbb{R} \to \mathbb{R}^3$$

is the derivative, $\forall\, t \in \mathbb{R}$,

$$\dot{\mathbf{x}}(t) = \frac{d}{dt}\mathbf{x}(t)$$

Acceleration $\ddot{\mathbf{x}}\colon \mathbb{R} \to \mathbb{R}^3$ is the second derivative,

$$\ddot{\mathbf{x}} = \frac{d^2}{dt^2}\mathbf{x}$$

Force on an object is $\mathbf{F}\colon \mathbb{R} \to \mathbb{R}^3$.

Newton's Second Law

Newton's second law states $\forall\, t \in \mathbb{R}$,

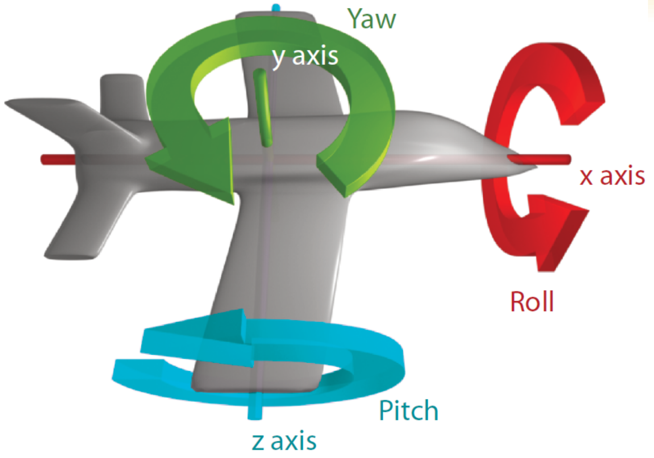
$$\mathbf{F}(t) = M\ddot{\mathbf{x}}(t)$$

where M is the mass. To account for initial position and velocity, convert this to an integral equation

$$\begin{aligned}\mathbf{x}(t) &= \mathbf{x}(0) + \int_0^t \dot{\mathbf{x}}(\tau)d\tau \\ &= \mathbf{x}(0) + t\dot{\mathbf{x}}(0) + \frac{1}{M}\int_0^t\int_0^\tau \mathbf{F}(\alpha)d\alpha d\tau,\end{aligned}$$

Orientation

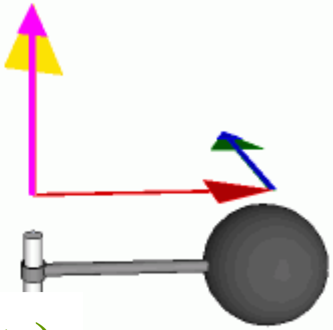
- Orientation: $\theta : \mathbb{R} \to \mathbb{R}^3$
- Angular velocity: $\dot{\theta} : \mathbb{R} \to \mathbb{R}^3$
- Angular acceleration: $\ddot{\theta} : \mathbb{R} \to \mathbb{R}^3$
- Torque: $\mathbf{T} : \mathbb{R} \to \mathbb{R}^3$



$$\theta(t) = \begin{bmatrix} \dot{\theta}_x(t) \\ \dot{\theta}_y(t) \\ \dot{\theta}_z(t) \end{bmatrix} = \begin{bmatrix} \text{roll} \\ \text{yaw} \\ \text{pitch} \end{bmatrix}$$

Torque: Angular version of Force

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



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

angular momentum, momentum

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 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.

If the object is spherical, this reluctance is the same around all axes, so it reduces to a constant scalar **I** (or equivalently, to a diagonal matrix I with equal diagonal elements *I*).

$$\mathbf{T}(t) = I\ddot{\theta}(t)$$

For a spherical object

Rotational velocity is the integral of acceleration,

$$\dot{\theta}(t) = \dot{\theta}(0) + \int_0^t \ddot{\theta}(\tau) d\tau,$$

$$\dot{\theta}(t) = \dot{\theta}(0) + \frac{1}{I} \int_0^t \mathbf{T}(\tau) d\tau.$$

Orientation is the integral of rotational velocity,

$$\begin{aligned} \theta(t) &= \theta(0) + \int_0^t \dot{\theta}(\tau) d\tau \\ &= \theta(0) + t\dot{\theta}(0) + \frac{1}{I} \int_0^t \int_0^\tau \mathbf{T}(\alpha) d\alpha d\tau \end{aligned}$$

Simplified Model

- Model-order Reduction

Yaw dynamics:

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

To account for initial angular velocity, write as

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

Simplified Model of Helicopter

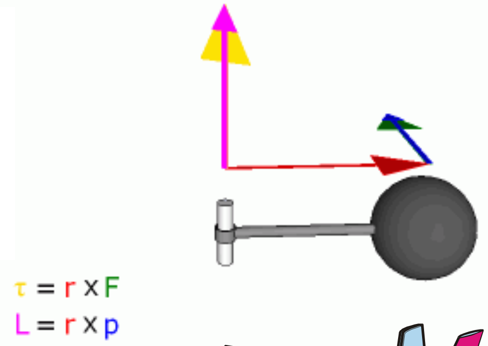
- the force produced by the tail rotor must counter the torque produced by the main rotor
- Assumptions:
  - helicopter position is fixed at the origin
  - helicopter remains vertical, so pitch and roll are fixed at zero
- the moment of inertia reduces to a scalar that represents a torque that resists changes in yaw

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

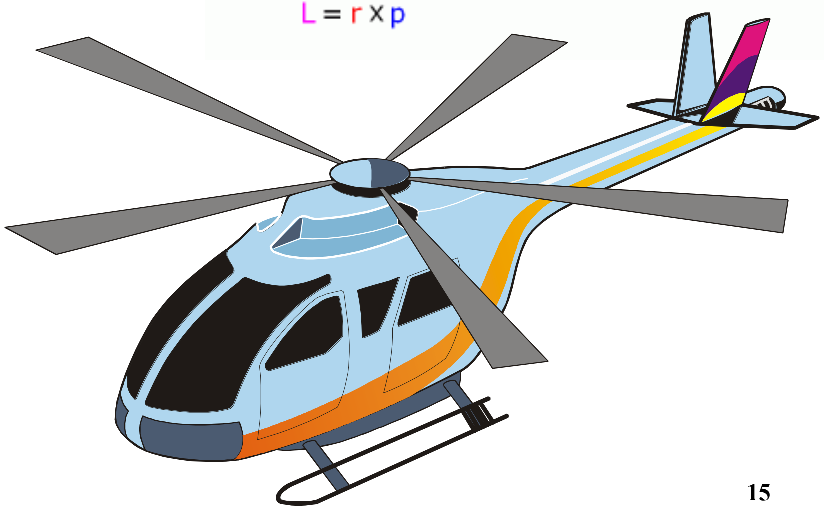
$$\dot{\theta}_y(t) = \dot{\theta}_y(0) + \frac{1}{I_{yy}} \int_0^t T_y(\tau)d\tau$$

Feedback Control Problem

A helicopter without a tail rotor, like the one below, will spin uncontrollably due to the torque induced by friction in the rotor shaft.



Control system problem:
Apply torque using the tail rotor to counterbalance the torque of the top rotor.

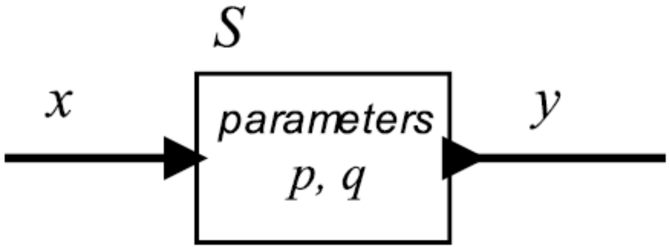


Actor Model

- Mathematical Model of Concurrent Computation
- Actor is an unit of computation
- Actors can
  - Create more actors
  - Send messages to other actors
  - Designate what to do with the next message
- Multiple actors may execute at the same time

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



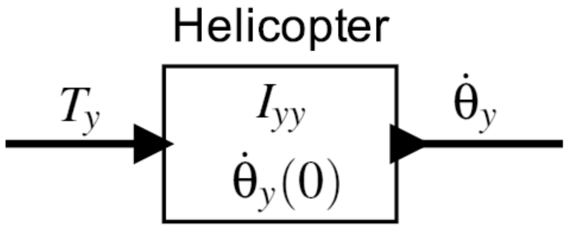
$$x\colon \mathbb{R} \to \mathbb{R}, \quad y\colon \mathbb{R} \to \mathbb{R}$$

$$S\colon X \to Y$$

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

Actor Model of the Helicopter

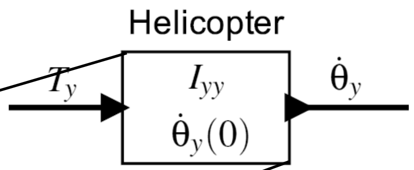
- Input is the net torque of the tail rotor and the top rotor. Output is the angular velocity around the y-axis.



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

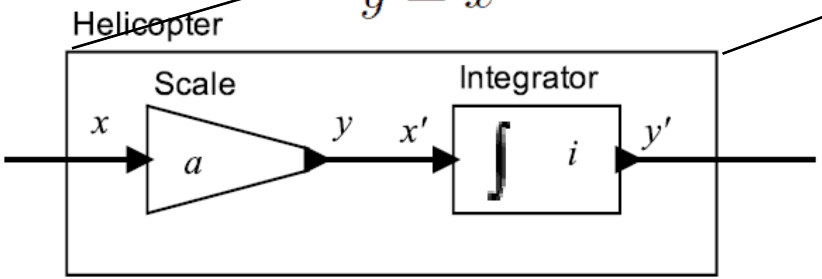
$$\dot{\theta}_y(t) = \dot{\theta}_y(0) + \frac{1}{I_{yy}} \int_0^t T_y(\tau) d\tau$$

Composition of Actor Model



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

$$y = x'$$



$$x = T_y \qquad y' = \dot{\theta}_y$$

$$\forall\, t \in \mathbb{R}, \quad y(t) = ax(t)$$

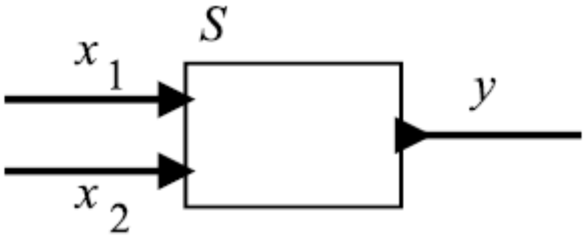
$$y'(t) = i + \int_0^t x'(\tau)d\tau$$

$$y = ax$$

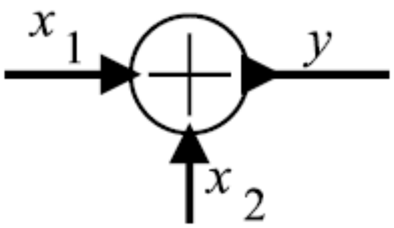
$$i = \dot{\theta}_y(0)$$

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

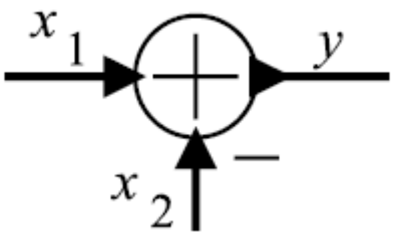
Actor Models with Multiple Inputs



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



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



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

Modern Actor Based Platforms

- Simulink (The MathWorks)
- Labview (National Instruments)
- Modelica (Linkoping)
- OPNET (Opnet Technologies)
- Polis & Metropolis (UC Berkeley)
- Gabriel, Ptolemy, and Ptolemy II (UC Berkeley)
- OCP, open control platform (Boeing)
- GME, actor-oriented meta-modeling (Vanderbilt)
- SPW, signal processing worksystem (Cadence)
- System studio (Synopsys)
- ROOM, real-time object-oriented modeling (Rational)
- Easy5 (Boeing)
- Port-based objects (U of Maryland)
- I/O automata (MIT)
- VHDL, Verilog, SystemC (Various)

Example LabVIEW Screenshot