IVR: Introduction to Control Theory

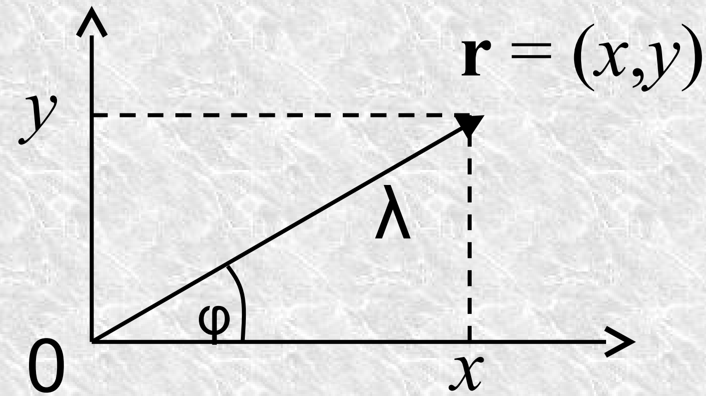
OVERVIEW

- Kinematics
- Differential equations describing robot movements

Kinematic (motion) models

- Differentiating the geometric model provides a motion model (hence sometimes these terms are used interchangeably)
- This may sometimes be a method for obtaining linearity (i.e. by looking at position change in the limit of very small changes)

Example:
A simple arm model



$$x = \lambda \cos \varphi$$

$$y = \lambda \sin \varphi$$

$$\varphi = \mathrm{atan2}(x,y)$$

$$y(t) = \lambda \sin \varphi(t)$$

e.g. vertical movement:

$$\frac{dy}{dt} = \lambda \cos \varphi(t) \frac{d\varphi}{dt}$$

Dynamic (kinetic) models

- Kinematic models neglect forces: motor torques, inertia, gravity, friction, …
- Forces arise according to Newtons 2nd law: $F = m\,a$
- $F = m\,a = m\,d^2y/dt^2$ (need to know mass!)

$$y(t) = \lambda \sin \varphi(t)$$

$$\frac{dy}{dt} = \lambda \cos \varphi(t) \frac{d\varphi}{dt}$$

$$\frac{d^2y}{dt^2} = -\lambda \sin \varphi(t)\,(d\varphi/dt)^2 + \lambda \cos \varphi(t)\, d^2\varphi/dt^2$$

- To control a system, we need to understand the continuous process

Dynamical systems

- Differ from standard computational view on systems:
  - **Perception-action loop** i.e. the influence of outputs onto inputs is taken into account (instead of: input → processing → output)
  - Set of all possible states forms a *state-space* (can be continuous or discrete)
  - Behaviour is a trajectory in the state space
- Not only physical systems: On-going debate whether human cognition is better described as computation or as a dynamical system (e.g. van Gelder, 1998). Similarly: society, economy, ecosystems etc.

Differential Equations

Using known relations between quantities and their rate of change in order to find out how these quantities change

- Mathematics:
  Equation that is to be solved for an unknown function
- Physics:
  Description of processes in nature
- Engineering:
  Realizability of a goal by a plant by including control terms
- Informatics:
  Tool for realistic modeling

$$\frac{dx(t)}{dt} = f\left(x(t)\right)$$

$$x(t_0) = x_0$$

Solving a simple diff-equation

$$\boxed{\dot{x}(t) = a \cdot x(t)}$$

$$\frac{dx(t)}{dt} = a \cdot x(t)$$

$$(x(t) \neq 0)$$

$$\frac{1}{x(t)} \frac{dx(t)}{dt} = a$$

$$\int_{t_0}^{t} \frac{1}{x(t')} \frac{dx(t')}{dt'} dt' = \int_{t_0}^{t} a\, dt'$$

$$\int_{x_0 = x(t_0)}^{x} \frac{1}{x'} dx' = \int_{t_0}^{t} a\, dt'$$

$$\log x' \Big|_{x_0}^{x} = at' \Big|_{t_0}^{t}$$

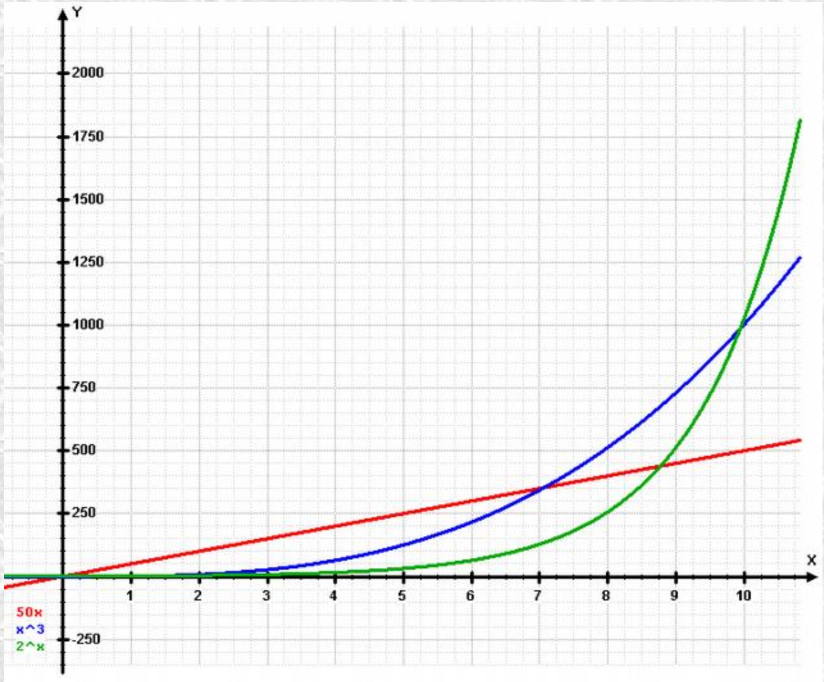
$$\log x - \log x_0 = at - at_0$$

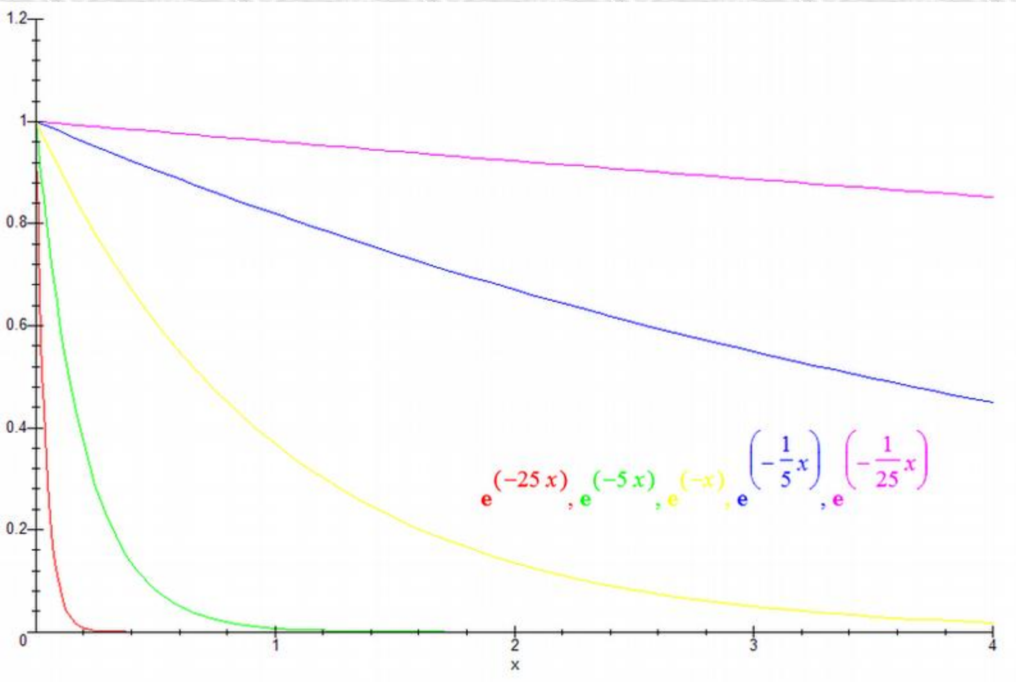
$$\log x(t) = \log x_0 + a(t - t_0)$$

$$x(t) = x_0 \cdot \exp a(t - t_0)$$

Differential Equations

$$\dot{x}(t) = a \cdot x(t) \quad \Longrightarrow \quad x(t) = x_0 \cdot \exp a(t - t_0)$$





$a > 0$ increases faster than any linear or polynomial function [for $x(0) > 0$]

$a < 0$ decay with time scale: $-1/a = \tau$

Problem: Robots are non-linear

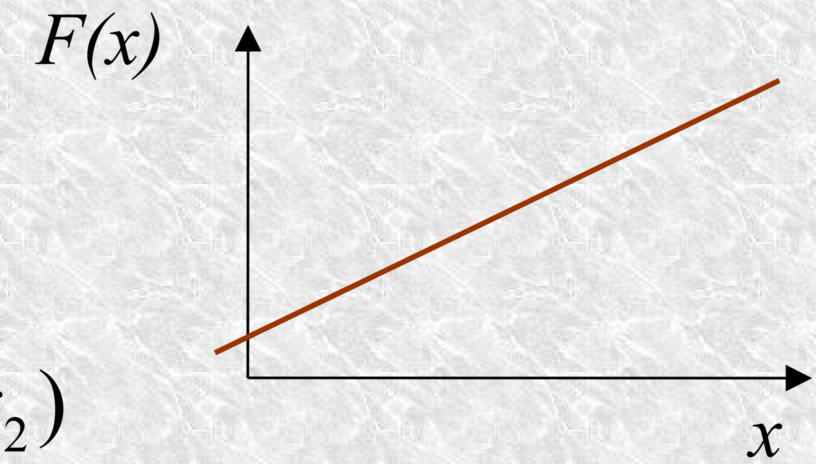
Linear differential equations (like the previous one) contain only linear combinations of variable function(s) and derivatives

- <u>Reminder:</u>

  Linear function:

  $$F(x) = a + bx$$

  $$F(x_1 + x_2) = F(x_1) + F(x_2)$$



- Linear combinations of solutions of the same linear equation are again solutions
  (in the previous differential equation different solutions differ only by x_0)
- For linear systems there are good formal methods
- However, dynamics of most robots is non-linear

Summary

- Whatever happens can be described as a dynamical system
- Differential equations are the native language of continuous dynamical systems
- Control systems describe the behaviour of robots
- For small step sizes, non-linear behaviour can often be linearly approximated
- In some cases we can use discretised form of the equations

Overview and Outlook

- Inverse and forward models (last time)
  describe transformations (flow of information) in a control system. Feedback control is less dependent on models as it uses plant outputs to determine control actions
- Modes of control (next lecture)
  - Open loop (based on an inverse model)
  - Feed-forward (open loop, but reacting pre-emptively to known disturbances)
  - Closed loop (feedback control)
- Feedback control (to be discussed later)
  - requires less domain knowledge
  - can use forward models to reduce delay effects