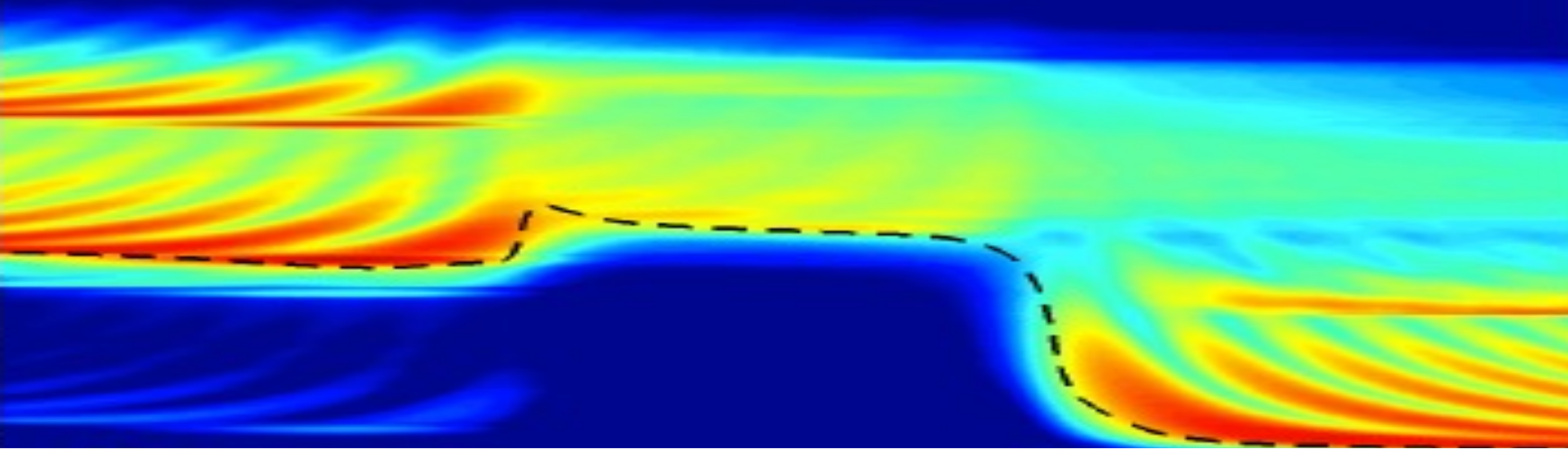


EECS 16B
Designing Information Devices and Systems II
Lecture 25

Prof. Yi Ma
Department of Electrical Engineering and Computer Sciences, UC Berkeley,
yima@eecs.berkeley.edu

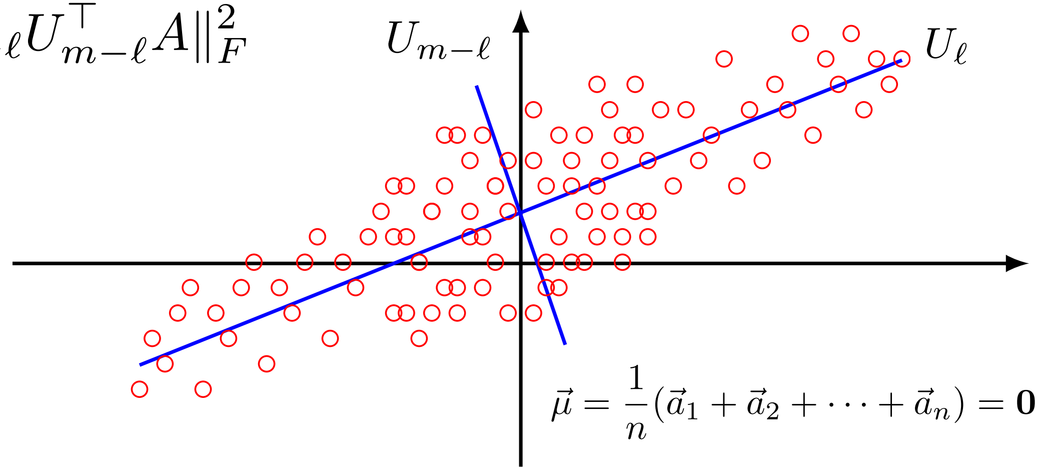
Outline

- Principal Component Analysis (Statistics)
  - Least Squares versus Principal Components
- Linearization of Nonlinear Systems

Principal Component Analysis (Statistics)

$U = [U_\ell, U_{m-\ell}] \in \mathbb{R}^{m \times m}$ orthogonal $\quad \|A\|_F^2 = \|UU^\top A\|_F^2 = \|U_\ell U_\ell^\top A\|_F^2 + \|U_{m-\ell} U_{m-\ell}^\top A\|_F^2$

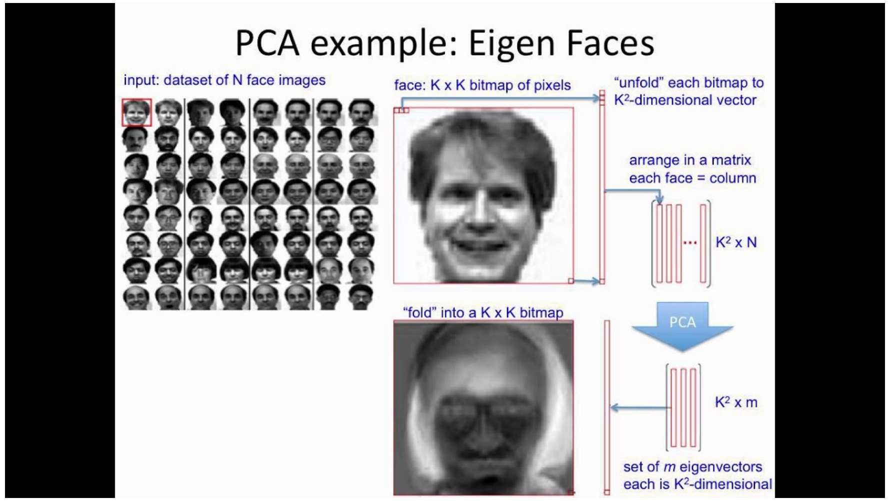
$$\max_{U_\ell} \|U_\ell U_\ell^\top A\|_F^2 \quad \Leftrightarrow \min_{U_\ell} \|A - U_\ell U_\ell^\top A\|_F^2 \quad \Leftrightarrow \quad \min_{U_{m-\ell}} \|U_{m-\ell} U_{m-\ell}^\top A\|_F^2$$

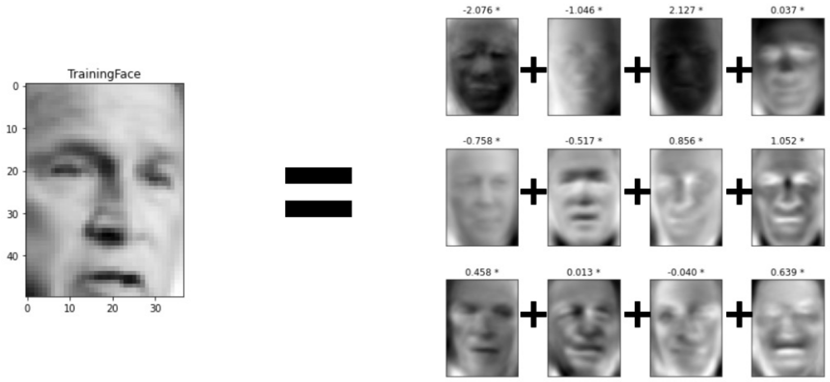


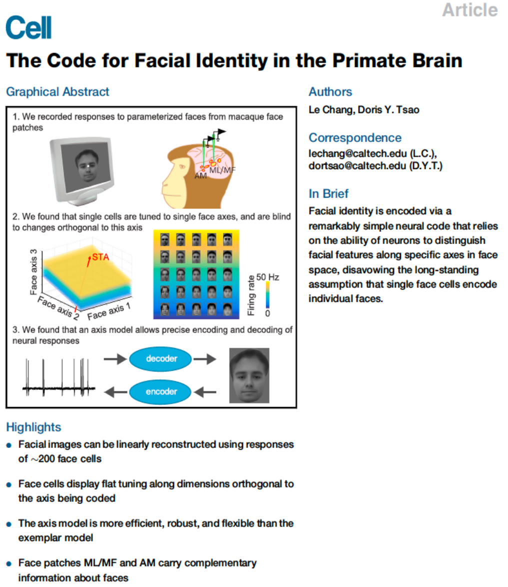
$\vec{\mu} = \frac{1}{n}(\vec{a}_1 + \vec{a}_2 + \cdots + \vec{a}_n) = \mathbf{0}$

Applications of PCA

- Eigenfaces [Turk & Pentland 1991]:







Article

Cell

The Code for Facial Identity in the Primate Brain

Graphical Abstract

1. We recorded responses to parameterized faces from macaque face patches
2. We found that single cells are tuned to single face axes, and are blind to changes orthogonal to this axis
3. We found that an axis model allows precise encoding and decoding of neural responses

Authors

Le Chang, Doris Y. Tsao

Correspondence

lechang@caltech.edu (L.C.),
dortsao@caltech.edu (D.Y.T.)

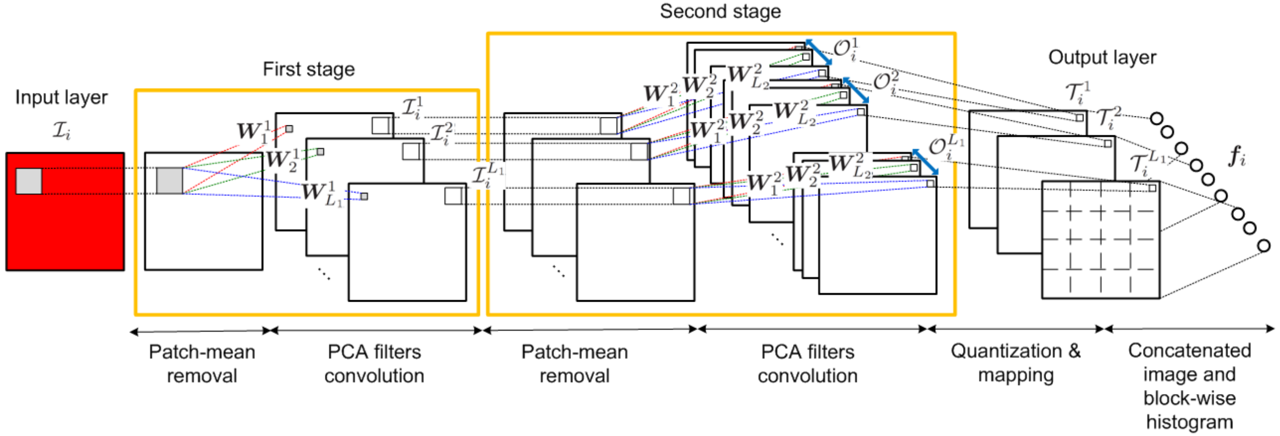
In Brief

Facial identity is encoded via a remarkably simple neural code that relies on the ability of neurons to distinguish facial features along specific axes in face space, disavowing the long-standing assumption that single face cells encode individual faces.

Highlights

- Facial images can be linearly reconstructed using responses of ~200 face cells
- Face cells display flat tuning along dimensions orthogonal to the axis being coded
- The axis model is more efficient, robust, and flexible than the exemplar model
- Face patches ML/MF and AM carry complementary information about faces

- PCANet [Chan & Ma et. al. 2015]:

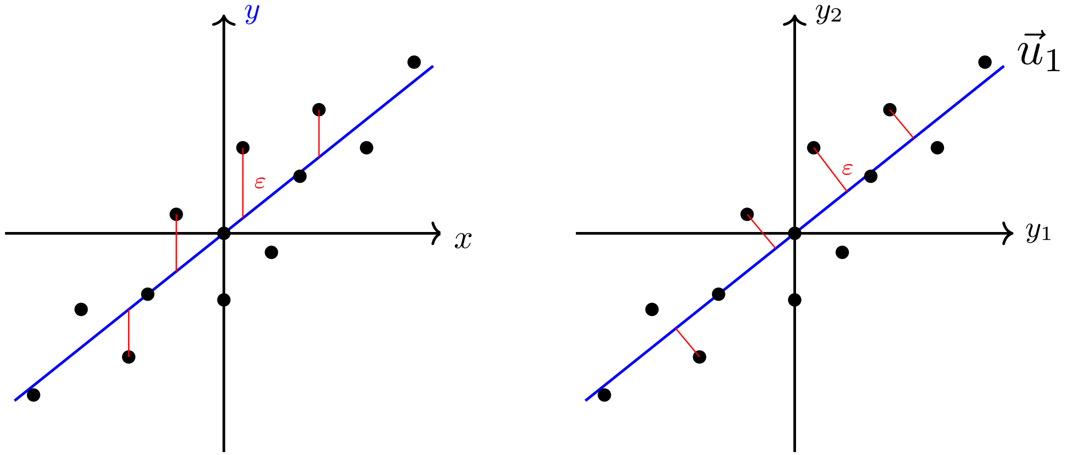


Recognition rates (%) on FERET dataset.

Probe sets	*Fb*	*Fc*	*Dup-I*	*Dup-II*	Avg.
LBP [18]	93.00	51.00	61.00	50.00	63.75
DMMA [25]	98.10	98.50	81.60	83.20	89.60
P-LBP [21]	98.00	98.00	90.00	85.00	92.75
POEM [26]	99.60	99.50	88.80	85.00	93.20
G-LQP [27]	99.90	100	93.20	91.00	96.03
LGBP-LGXP [28]	99.00	99.00	94.00	93.00	96.25
sPOEM+POD [29]	99.70	100	94.90	94.00	97.15
GOM [30]	99.90	100	95.70	93.10	97.18
PCANet-1 (Trn. CD)	99.33	99.48	88.92	84.19	92.98
PCANet-2 (Trn. CD)	99.67	99.48	95.84	94.02	97.25
PCANet-1	99.50	98.97	89.89	86.75	93.78
PCANet-2	99.58	100	95.43	94.02	97.26

Least Squares (Regression) versus PCA

Prediction versus **Correlation**

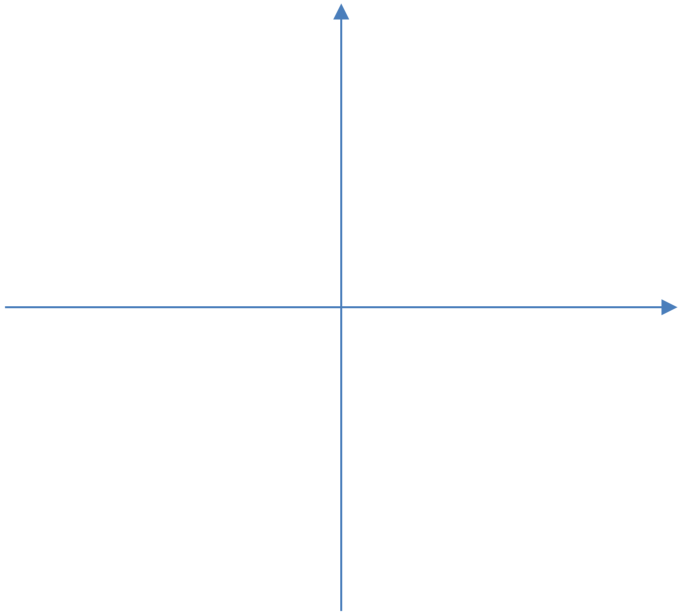


Supervised versus **Unsupervised**

Least Squares (Regression) versus PCA

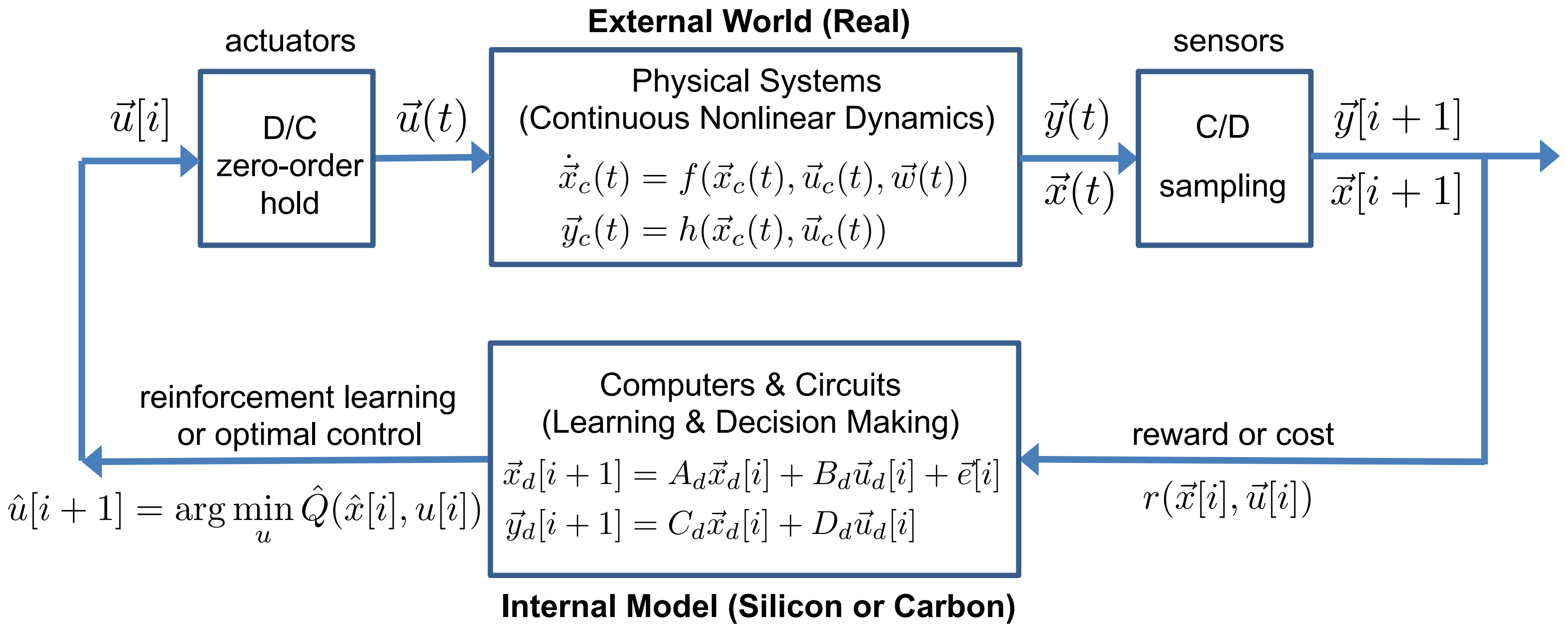
Example: $A = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 1 & 1 \end{bmatrix}$

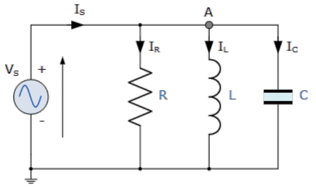
Prediction versus Correlation

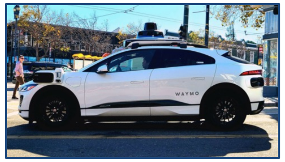


System Modeling & Control

All autonomous intelligent (AI) systems rely on closed-loop learning and control:







mathematical modeling
from first principles

$$\dot{\vec{x}}_c(t) = f(\vec{x}_c(t), \vec{u}_c(t), \vec{w}(t))$$
$$\vec{y}_c(t) = h(\vec{x}_c(t), \vec{u}_c(t))$$

approximation
& linearization

$$\dot{\vec{x}}(t) = A\vec{x}(t) + B\vec{u}(t) + \vec{n}(t)$$
$$\vec{y}(t) = C\vec{x}(t) + D\vec{u}(t)$$

discretization
& digitization

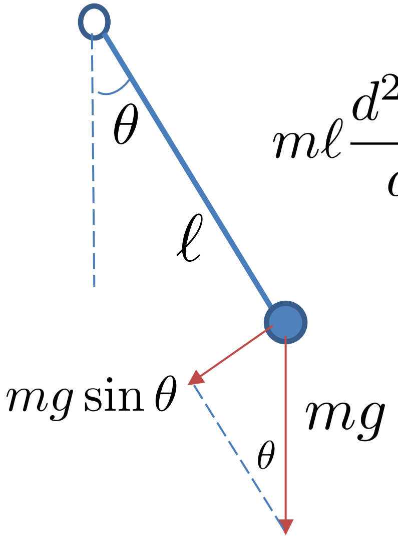
$$\vec{x}_d[i+1] = A_d\vec{x}_d[i] + B_d\vec{u}_d[i] + \vec{e}[i]$$
$$\vec{y}_d[i+1] = C_d\vec{x}_d[i] + D_d\vec{u}_d[i]$$

Linear versus Nonlinear Systems

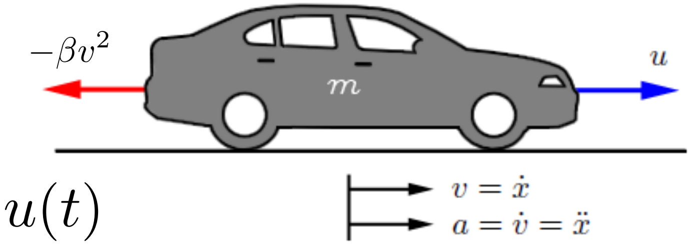
Objectives: Identification (learning), Analysis (stability), Control (closed-loop feedback)

	Continuous Time	Discrete Time
Linear Control Systems	$\frac{d\vec{x}(t)}{dt} = A\vec{x}(t) + B\vec{u}(t)$	$\vec{x}[i+1] = A\vec{x}[i] + B\vec{u}[i]$
Nonlinear Control Systems	$\frac{d\vec{x}(t)}{dt} = \vec{f}(\vec{x}(t), \vec{u}(t))$	$\vec{x}[i+1] = \vec{f}(\vec{x}[i], \vec{u}[i])$
Autonomous Systems	$\frac{d\vec{x}(t)}{dt} = \vec{f}(\vec{x}(t))$	$\vec{x}[i+1] = \vec{f}(\vec{x}[i])$

Nonlinear Systems: Examples



$$m\ell\frac{d^2\theta(t)}{dt^2} = -k\ell\frac{d\theta(t)}{dt} - mg\sin\theta(t)$$



$$m\frac{v(t)}{dt} = -\beta v(t)^2 + u(t)$$

Nonlinear Autonomous Systems: Equilibrium Points

$$\frac{d\vec{x}(t)}{dt} = \vec{f}(\vec{x}(t)) \ \in \mathbb{R}^n$$

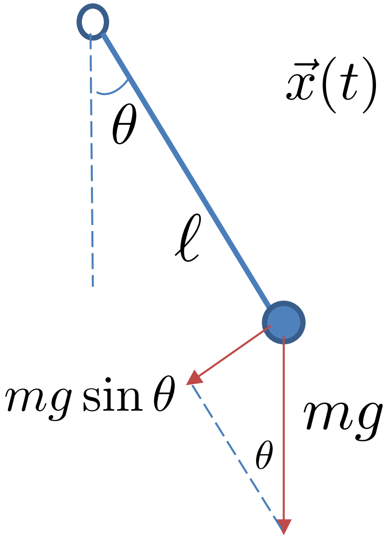
$$\vec{x}[i+1] = \vec{f}(\vec{x}[i]) \ \in \mathbb{R}^n$$

Nonlinear Autonomous Systems: Linearization

Scalar case: $\frac{dx(t)}{dt} = f(x(t))$

Vector case: $\frac{d\vec{x}(t)}{dt} = \vec{f}(\vec{x}(t)) \in \mathbb{R}^n$

Nonlinear Autonomous Systems: Example

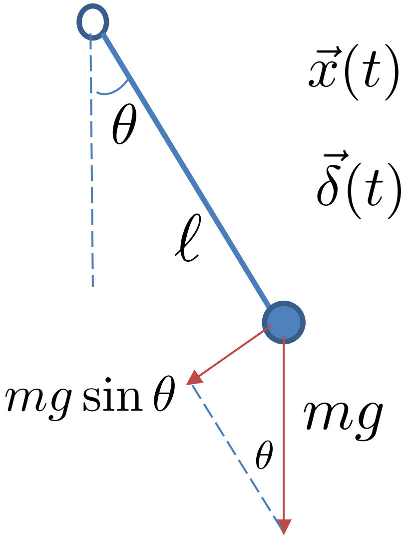


$$\vec{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} \theta(t) \\ \frac{d\theta(t)}{dt} \end{bmatrix}$$

$$\frac{dx_1(t)}{dt} = x_2(t) \qquad = f_1(x_1(t), x_2(t))$$

$$\frac{dx_2(t)}{dt} = -\frac{g}{\ell}\sin(x_1(t)) - \frac{k}{m}x_2(t) \quad = f_2(x_1(t), x_2(t))$$

Nonlinear Autonomous Systems: Example



$$\vec{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} \theta(t) \\ \frac{d\theta(t)}{dt} \end{bmatrix}$$

$$\vec{\delta}(t) = \vec{x}(t) - \vec{x}^*$$

$$\frac{d\vec{\delta}(t)}{dt} = \frac{d\vec{x}(t)}{dt} = \vec{f}(\vec{x}(t))$$

$$J_{\vec{x}}\vec{f}(\vec{x}) \mid_{(x_1^*, x_2^*)} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(x_1^*, x_2^*) & \frac{\partial f_1}{\partial x_2}(x_1^*, x_2^*) \\ \frac{\partial f_2}{\partial x_1}(x_1^*, x_2^*) & \frac{\partial f_2}{\partial x_2}(x_1^*, x_2^*) \end{bmatrix}$$

Nonlinear Control Systems: Operating Points

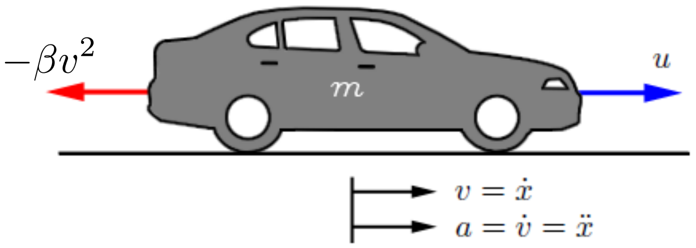
$$\frac{d\vec{x}(t)}{dt} = \vec{f}(\vec{x}(t), \vec{u}(t)) \ \in \mathbb{R}^n$$

$$\vec{x}[i+1] = \vec{f}(\vec{x}[i], \vec{u}[i]) \ \in \mathbb{R}^n$$

Nonlinear Control Systems: Linearization

Scalar case: $$\frac{dx(t)}{dt} = f(x(t), u(t))$$

Nonlinear Control Systems: Example



$$\frac{dx(t)}{dt} = -\frac{\beta}{m}x(t)^2 + \frac{1}{m}u(t) = f(x(t), u(t))$$

$$m\frac{v(t)}{dt} = -\beta v(t)^2 + u(t)$$