Learning and Adaptive Control for Robots Course

Overview & Exam Preparation

Exam Format

The exam lasts a total of 25 minutes:

- Upon entering the room, you pick at random 1 question.

- You present your answers on the black board.

- The exam will also consist of a discussion over the topic of the question and you will be asked to answer questions from the jury.

Exam is closed book but you can bring one A4 recto-verso page with *personal* ***handwritten*** notes.

Exam Material

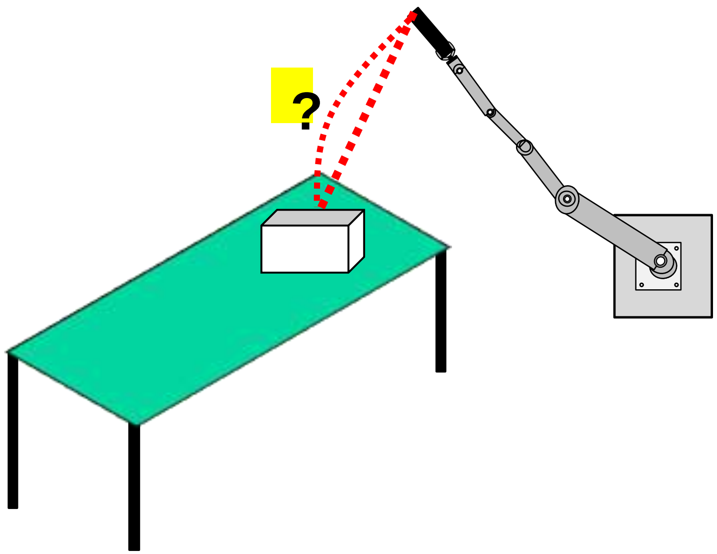
Today's overview highlights only some key components of each technique seens in class. The exam can cover any of the topic we have seen in:

- Slides & videos
- Material in the associated Book chapters
- Solutions to the pen and paper exercises
- Material done during the matlab and robotic practice sessions

Overview

Motivation for Use of DS

- Real-time adaptation to disturbances
- Closed-form expression
- Embed a flow of trajectories, all of which guaranteed to reach the target



Mathematical Expression

DS control law (1st order ordinary differential equation)

$$\dot{x} = f(x)$$

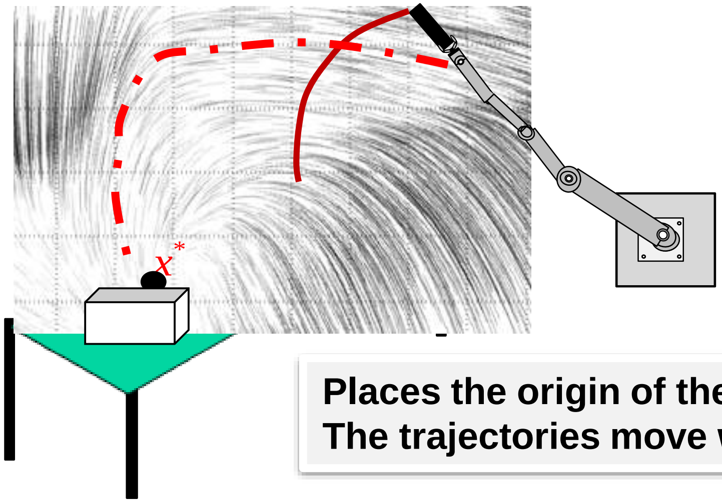
$x \in \mathbb{R}^N$: Robot's state

$\dot{x} \in \mathbb{R}^N$: Time-derivative of state, velocity

The system is asymptotically stable at a target, *x, and only at the goal:**

$$\lim_{t \to \infty} f(x^*) = 0$$

$x^* = 0$



Places the origin of the system on the attractor.
The trajectories move with the origin.

Learning DS: SEDS

Generate an estimate of the DS through Gaussian Mixture Regression:

$$\dot{x} = f\left(x;\left\{A^k, b^k\right\}_{k=1}^{K}\right) := \sum_{k=1}^{K} \gamma_k(x)\left(A^k + b^k\right): \quad \text{Mixture of } K \text{ linear DS}$$

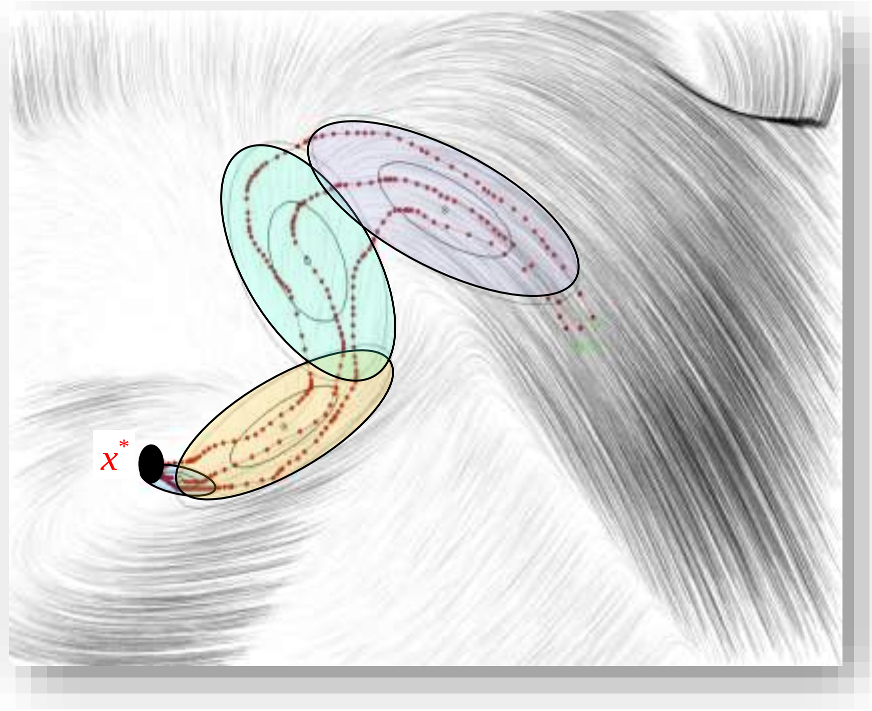
Learn parameters of the Gauss function as a constrained optimization problem:

Two possible objective functions:
1) Maximum likelihood
2) Mean-square error

Under several constraints, among which:

a) $b^k = -A^k x^*$ - Stability at attractor

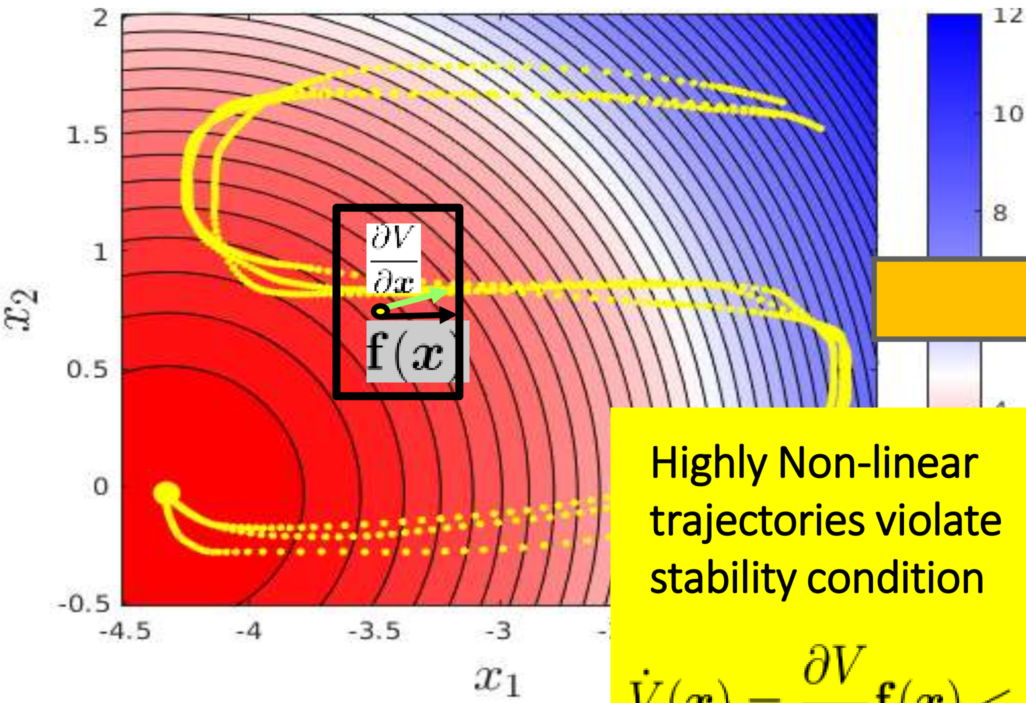
b) $A^k + \left(A^k\right)^T \prec 0 \quad \forall k$ - Energy decreases



Learning DS: LPV - DS

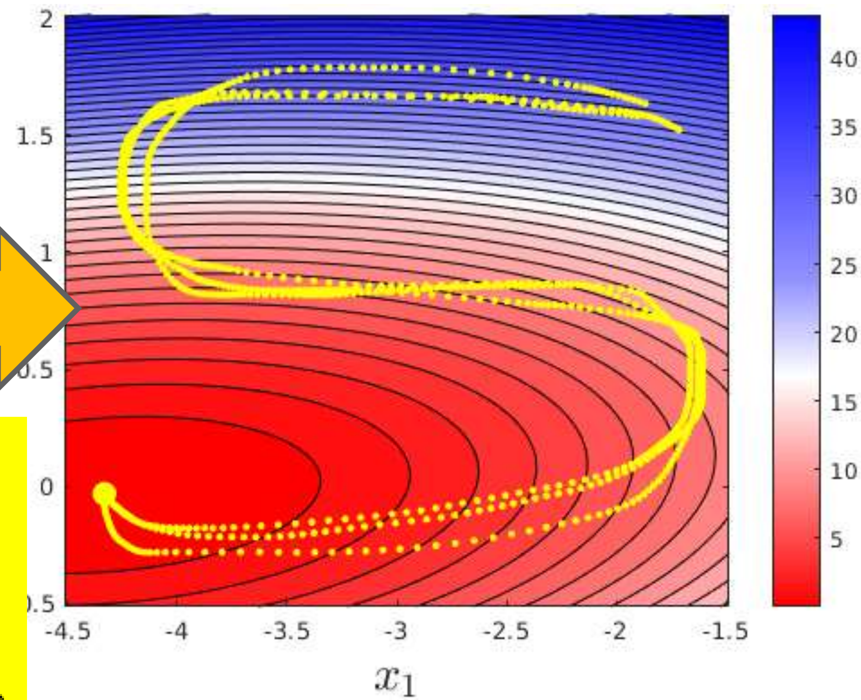
Relax isotropic form of the Lyapunov function

$$V(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{x}^*)^T(\boldsymbol{x} - \boldsymbol{x}^*)$$



Parameterized Quadratic Lyapunov Function **(P-QLF)**

$$V(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{x}^*)^T \mathbf{P}(\boldsymbol{x} - \boldsymbol{x}^*)$$



Highly Non-linear trajectories violate stability condition

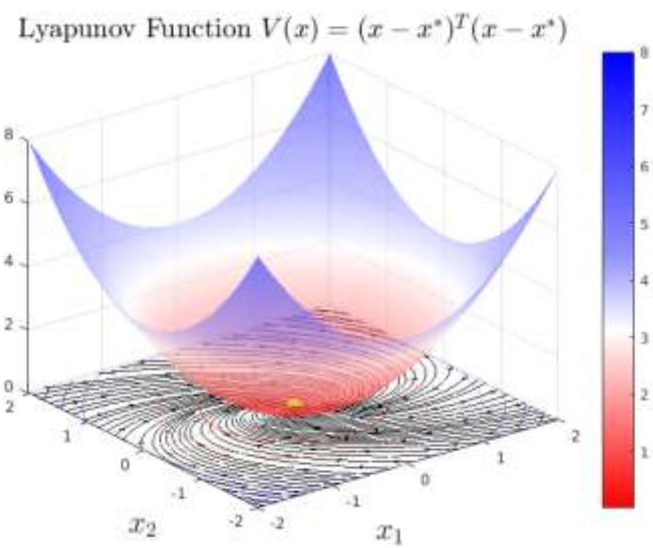
$$\dot{V}(\boldsymbol{x}) = \frac{\partial V}{\partial \boldsymbol{x}} \mathrm{f}(\boldsymbol{x}) < 0$$

If V is too conservative.

Learning DS: LPV - DS

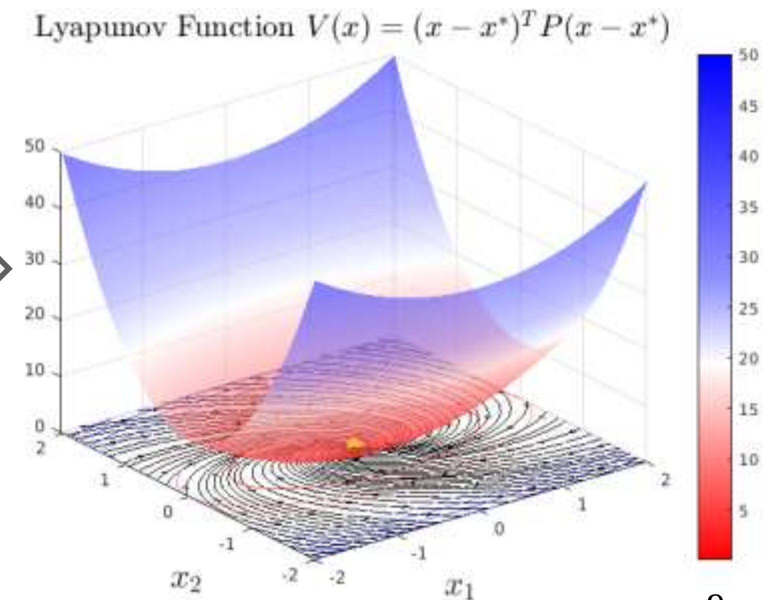
Relax isotropic form of the Lyapunov function

$$V(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{x}^*)^T(\boldsymbol{x} - \boldsymbol{x}^*)$$



Parameterized Quadratic Lyapunov Function (**P-QLF**)

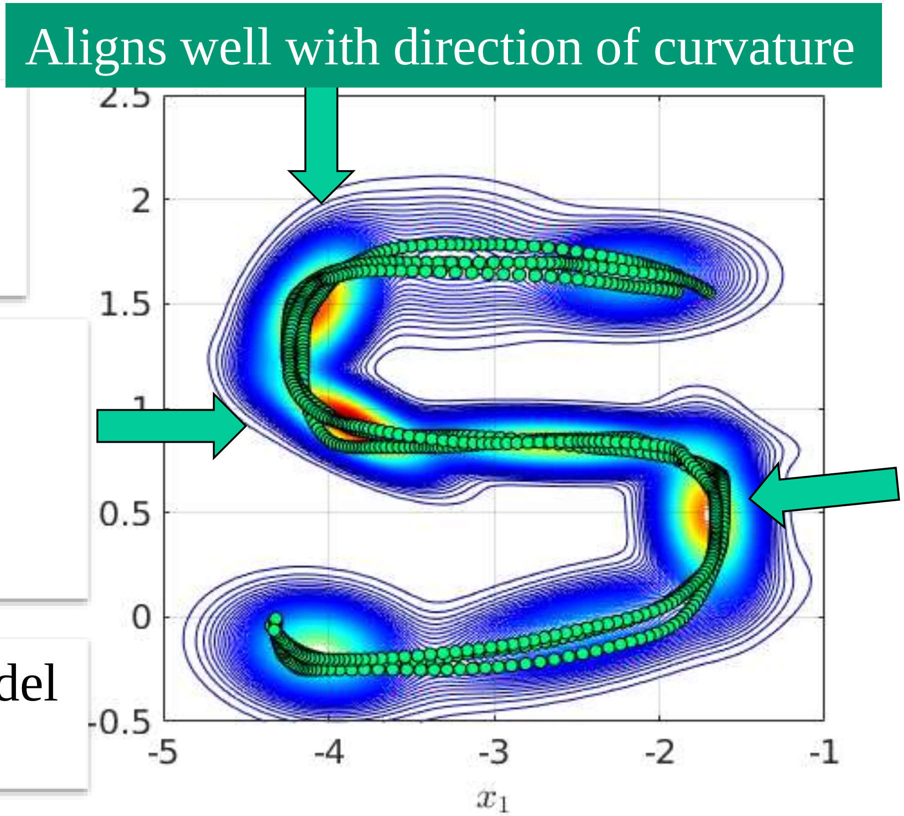
$$V(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{x}^*)^T\mathbf{P}(\boldsymbol{x} - \boldsymbol{x}^*)$$



9

Learning DS: LPV - DS

Devise a procedure to place the Gauss functions of the GMM, so as to follow the direction of motion of the data.



Introduce a new metric to cluster points. Points must be:
- Close to one another
- Have closely co-aligned velocities

Use Bayesian non-parametric Mixture Model training method (train only on position)

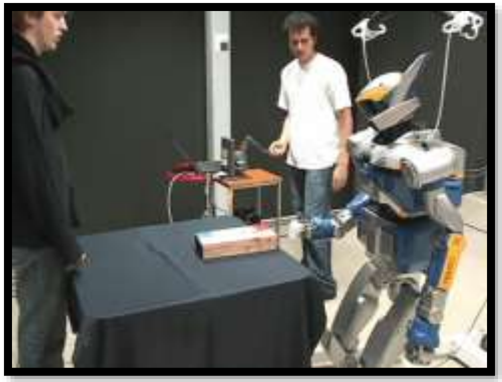
Once the GMM has been trained, update cross-covariance matrices to predict dynamics while satisfying stability constraints.

How do we gather data for learning?

Method to generate the data	Online mode	Need model of robot or world	Trainer	Number of training examples
Learning from human demonstrations	YES	NO	Anyone	<20
Optimal control	NO	YES	Skilled programmer	>100
RL (live)	NO	YES (model-based RL) NO (model-free RL)	Anyone (reward)	>100
RL (simulation)	YES	YES	Skilled programmer	>1,000

Interfaces to provide demonstrations

Teleoperation

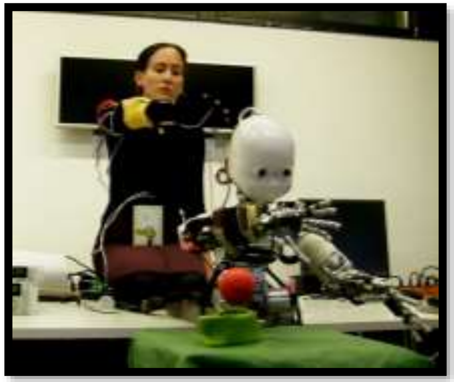


- Graphical user interface/Tablet
- Joysticks
- Exoskeleton
- Haptic devices

Kinesthetic Teaching

- Embody the robot
- Solves part of the correspondence problem (kinematic feasibility)
- Feel the interaction forces

Observational Learning



- Track human motion with video or motion sensors
- Natural demonstrations
- No force measurements

(must be inferred from motion)

Modulating a DS

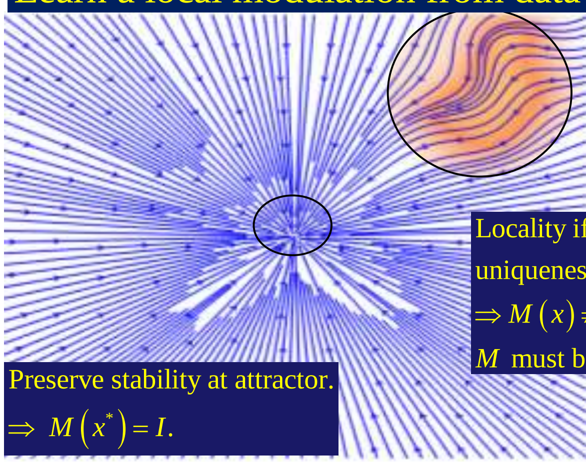
$$\dot{x} = M(x) f(x), \quad M(x) \in \mathbb{R}^{N \times N}$$

$$M(x) = (1 + \kappa(x)) R(x)$$

Modulates speed

Rotation $R \in \mathbb{R}^{N \times N}$

Learn a local modulation from data



Modulation depending on external input

$\dot{x} = M(x, s) f(x), \quad s \in \mathbb{R}^M$: external input

State and input–dependent scaling and rotation

$$M(x, s) = (1 + \kappa(x, s)) R(x, s)$$

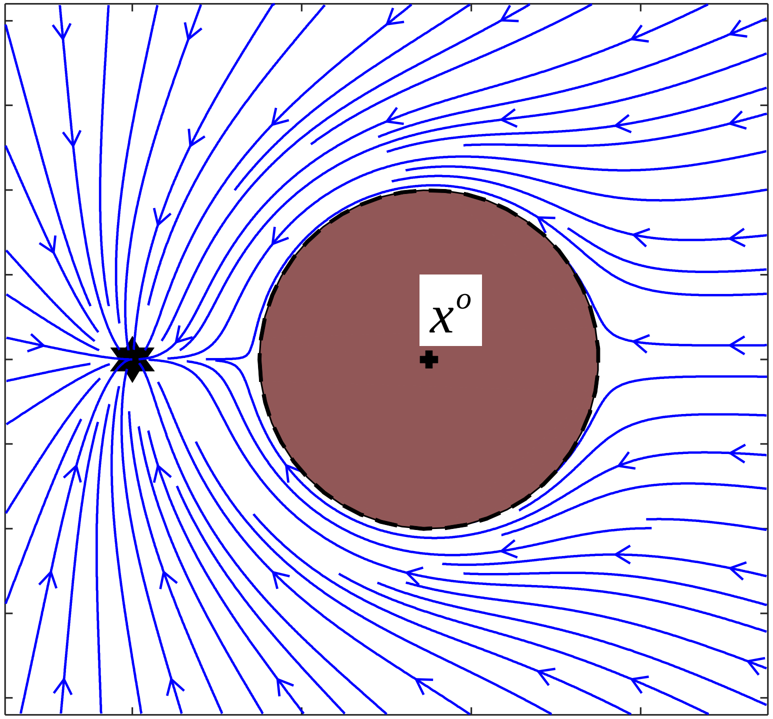
Modulating a DS – Obstacle Avoidance

$$\dot{x} = M(x)\, f(x), \quad M(x) \in \mathbb{R}^{N \times N}$$

$$M(x - x^o) = E(x - x^o)\, D(x - x^o)\, E(x - x^o)^{-1}.$$

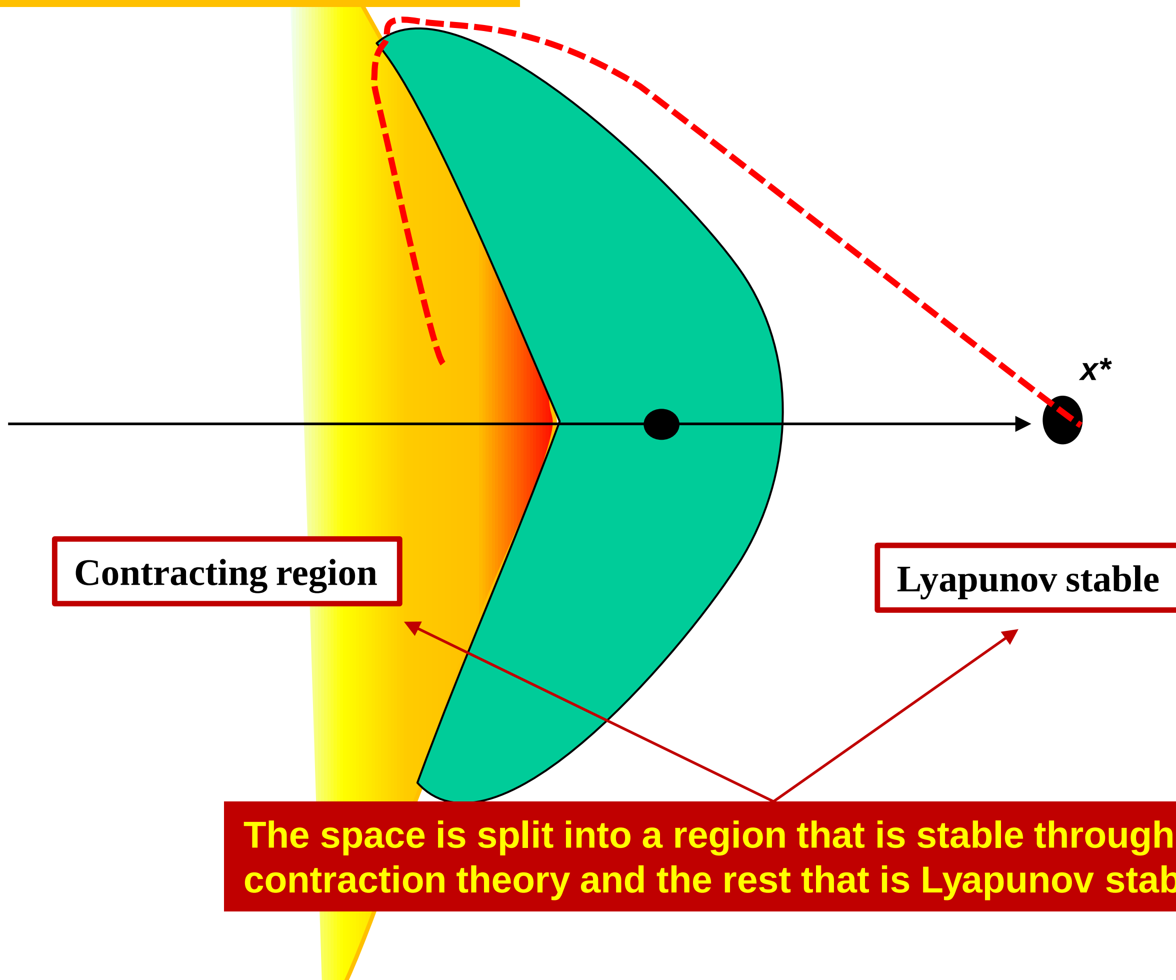
Construct a deflection through *E*.

Modulate deflection through eigenvalues *D*.



Modulating a DS – Obstacle Avoidance

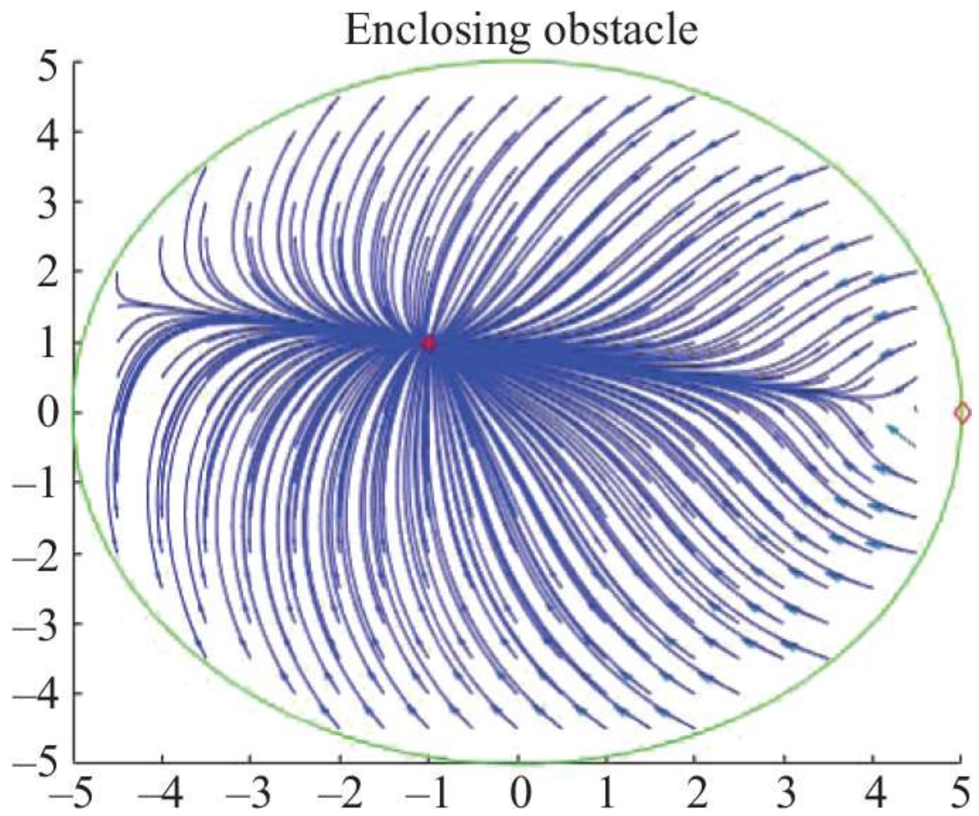
Extend the principle to enable obstacle avoidance of concave objects using contraction theory.



The space is split into a region that is stable through contraction theory and the rest that is Lyapunov stable.

Modulating a DS – Obstacle Avoidance

Flow is trapped inside the obstacle. This can be used to enclose DS in a given volume.

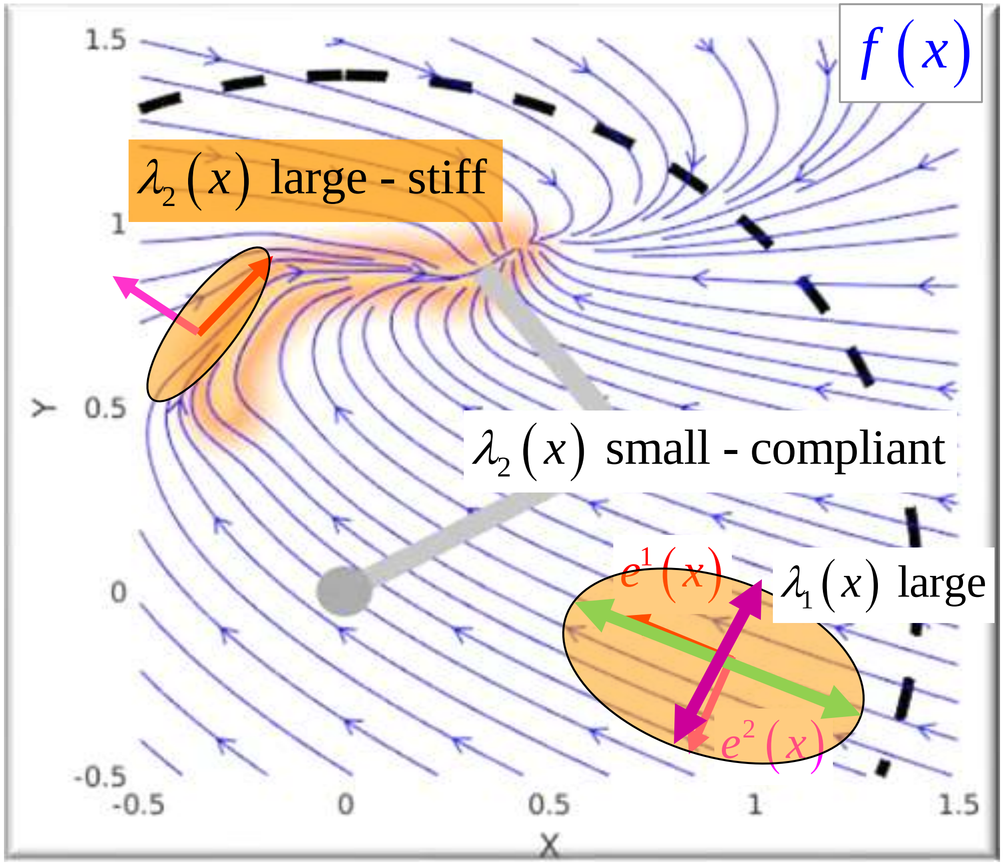


Impedance Control with DS

$$\tau_c = -D(x)(\dot{x} - f(x))$$

The eigenvalues set the impedance

$$D(x) = \begin{bmatrix} \lambda_1(x) & \\ & \lambda_2(x) \end{bmatrix}$$



Impedance Control with DS - Passivity

Passivity analysis

The system must remain passive under external disturbances τ_e.

We set: $\begin{cases} u = \tau_e \\ y = \dot{x} \end{cases}$

We define the storage function as W.

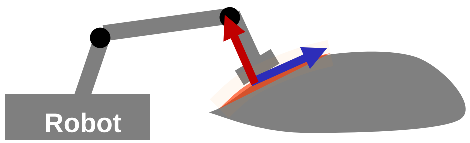
We verify that : $\dot{W} \leq \tau_e^{\ T} \dot{x}$

We set the storage function: $W = \underbrace{\frac{1}{2}\dot{x}^T M(x)\dot{x}}_{\text{Kinetic Energy}} + \underbrace{\lambda_1 V_f(x)}_{\text{Potential Energy of } f(x)}$

If f is Lyapunov stable, the system is passive.

Otherwise, revert to tank-based approach.

Force Control with DS



$$\tau_c = -D(x)(\dot{x} - \dot{x}_d)$$

Robot's control torques

$$\dot{x}_d = f(x) + f_n(x)$$

Reach/Move on the surface

Apply the contact force

To separate control of force and control of motion , we decompose the nominal DS into two components:

$$\dot{x}_d = f(x) + f_n(x) \qquad f_n(x) = 0 \quad (\text{in free space})$$

Preparation for the Exam

Theory:

You should be able to explain mathematically and in words (+ with schematics):

- Fundamental concepts of DS, such as stability under Lyapunov & Contraction Theory, asymptotical/global and local stability, passivity, definition of linear/nonlinear DS, limit cycle, saddle points, impedance control.
- Key steps of each DS algorithms seen in class (optimization approach to SEDS / LPV-DS; types of modulation and machine learning method used to estimate these; principle of impedance/force control with DS) and mathematical principles behind their theoretical guarantees

Exercises:

You should be able to solve the exercises done in class (or variants on these).

Practice sessions:

Examples of the dynamics generated by each algorithm; examples of the algorithm's sensitivity to certain choice of hyperparameters.

Preparation for the Exam

Role play with a friend!

One of you is the professor and the other the student.

As student, explain to your friend one technique.

A professor, ask questions to the student to test understanding of the technique, ask for examples, ask for justification of some statement (e.g. why is it stable?)