PART II

COUPLING AND MODULATING CONTROLLERS

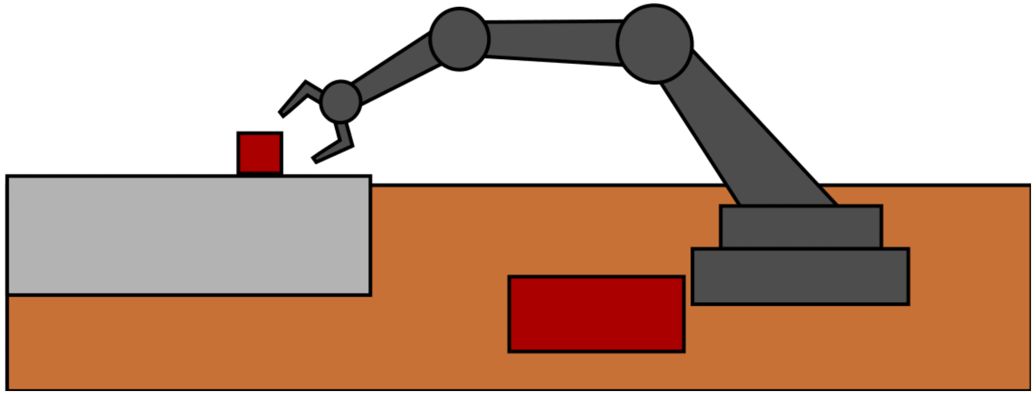
Chapter 9
Obstacle Avoidance with Dynamical Systems

Assumptions & Desiderata

Assume that we have, for starters, a nominal DS of the form: $\dot{x} = f(x)$

$$\& \quad \dot{x}^* = f(x^*) = 0$$

Modulate to enable real-time obstacle avoidance from one or multiple moving obstacles.



Recall: Main Properties for the Modulation

The modulated dynamics is given by:

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

Constrain $M(x)$ such that it preserves the following properties:

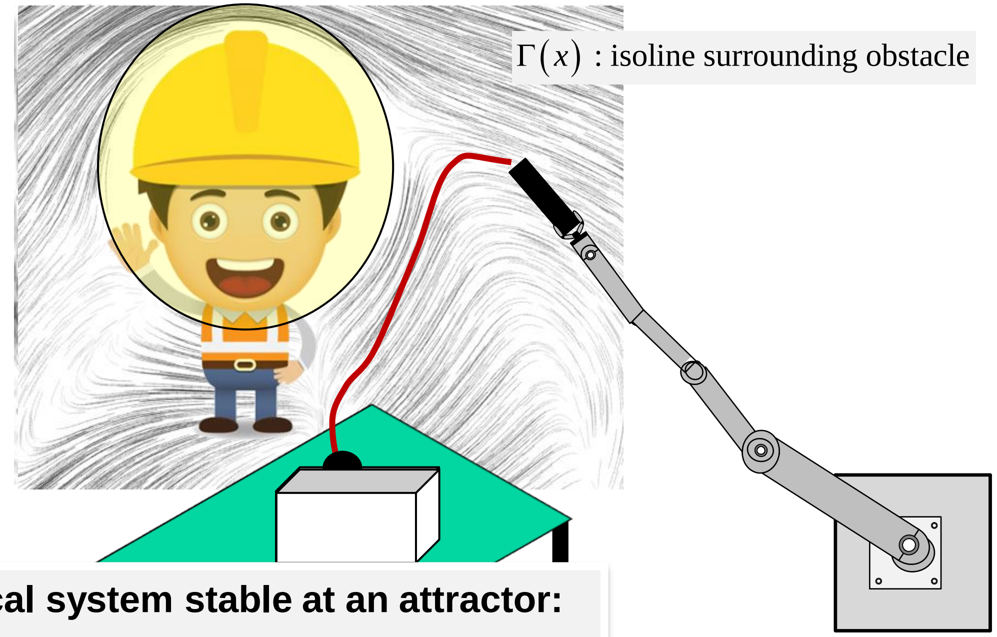
– Stability of the original attractor:

$\Rightarrow \exists$ a compact region $B(x, x^{*})$ around x^{*} where $M(x) = I, \forall x \in B(x, x^{*})$.

If $f(x) = Ax$, sufficient to have that $M(x) f(x) \prec 0, \forall x \in B(x, x^{*})$

– Uniqueness of the attractor: $\Rightarrow M(x) \neq 0, \; \forall x \neq x^{*}$. M must be full rank.

Obstacle avoidance with DS



Starts with an initial dynamical system stable at an attractor:

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

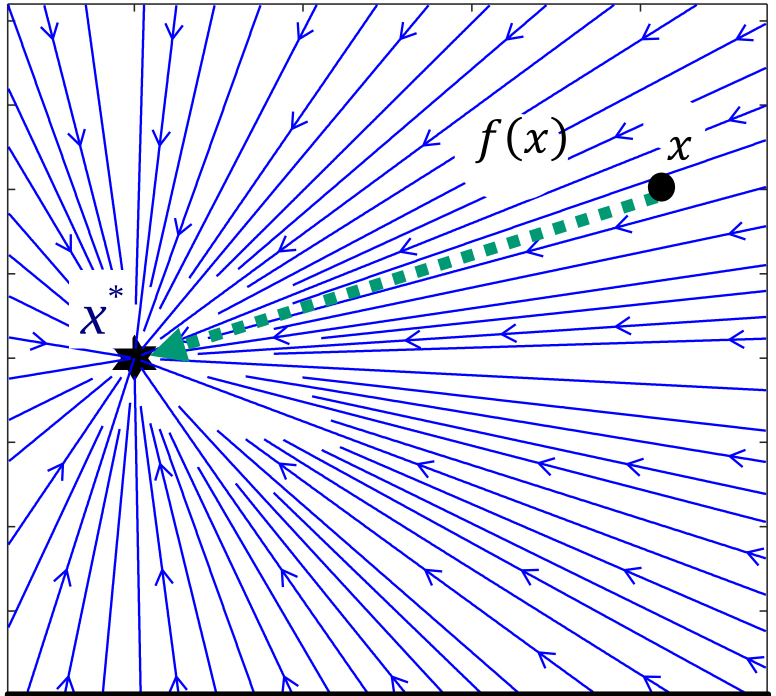
Add a modulation around the obstacle:

$$\dot{x} = M(\Gamma(x))\, f(x)$$

Guarantees that the robot will never penetrate the obstacle.
Guarantees that the robot will reach the goal.

Nominal Dynamical System

General formulation $\dot{x} = f\left(x\right)$ & $\dot{x}^* = f\left(x^*\right) = 0$



Initial dynamics with single attractor

Obstacle Avoidance Modulation Matrix

Modulation is given by: $\dot{x} = M(x)\, f(x)$

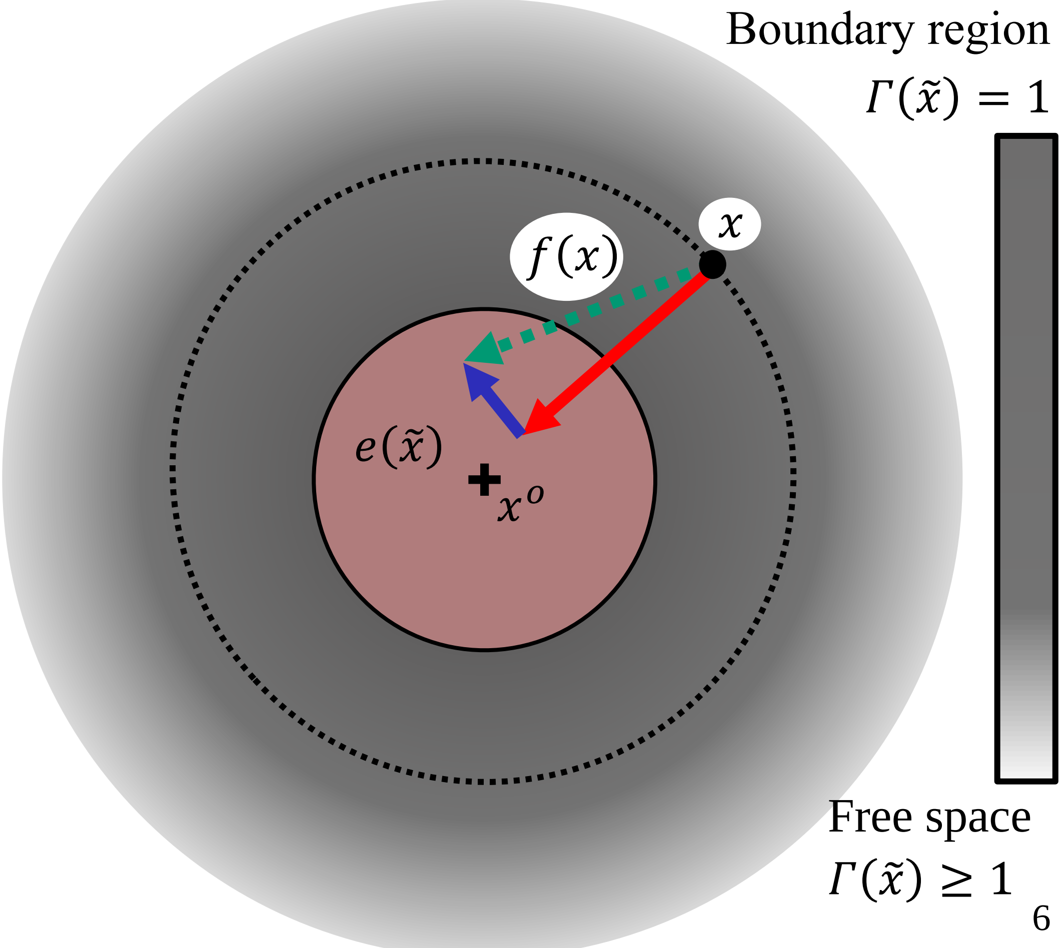
$$M(\tilde{x}) = E(\tilde{x})\, D(\tilde{x})\, E(\tilde{x})^{-1}$$

$$E(\tilde{x}) = \left[\, n(\tilde{x}) \quad e_1(\tilde{x}) \quad \ldots\ldots \quad e_{d-1}(\tilde{x}) \,\right]$$

Decomposition into normal $n(\tilde{x})$ & tangents $e_i(\tilde{x})$

Legend

$\tilde{x} \in \mathbb{R}^d$	Robot's relative state ($\tilde{x} = x - x^o$)
$E(\tilde{x})$	Decomposition matrix
$D(\tilde{x})$	Eigenvalue matrix
$n(\tilde{x})$	Normal to obstacle
$e_i(\tilde{x})$	Tangent to obstacle
$\Gamma(\tilde{x})$	Distance Function



Legend

$\tilde{x} \in \mathbb{R}^d$	Robot's relative state ($\tilde{x} = x - x^o$)
$E(\tilde{x})$	Decomposition matrix
$D(\tilde{x})$	Eigenvalue matrix
$n(\tilde{x})$	Normal to obstacle
$e_i(\tilde{x})$	Tangent to obstacle
$\Gamma(\tilde{x})$	Distance Function
$\lambda(\tilde{x})$	Eigenvalues

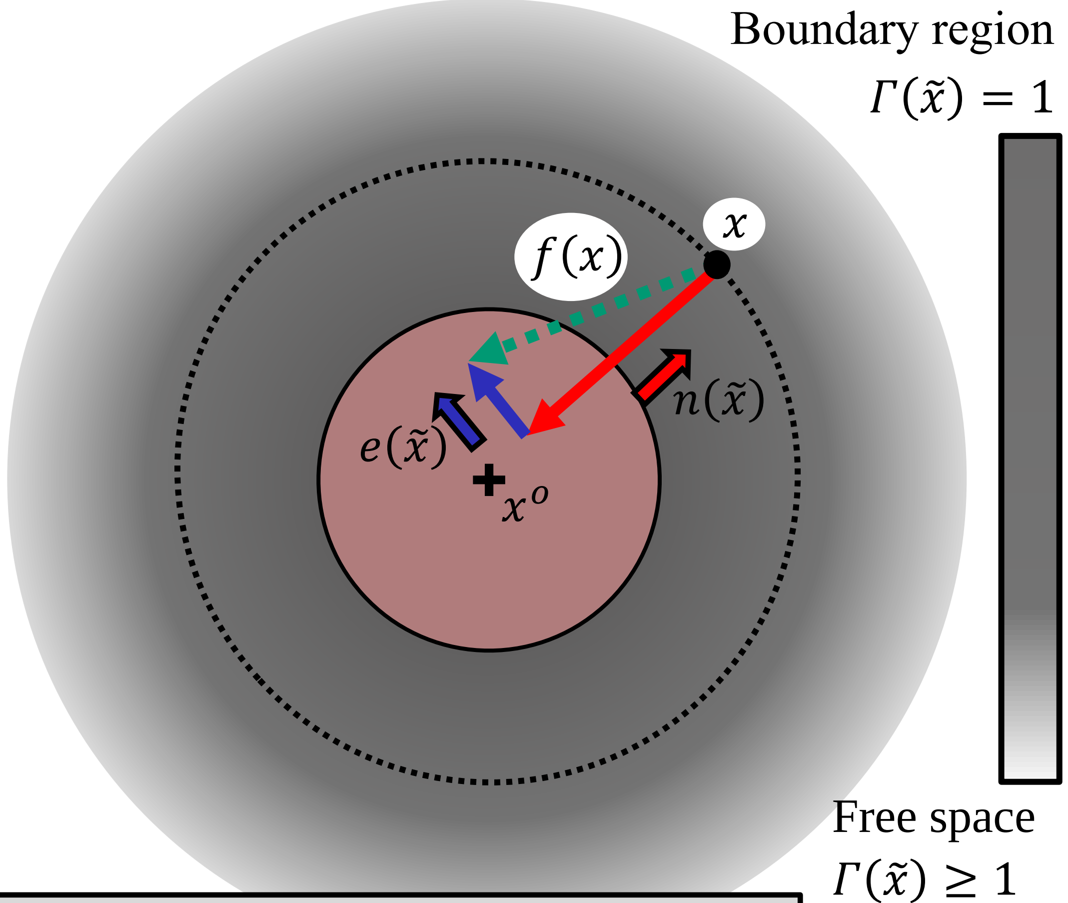
Modulation is given by: $\dot{x} = M(x) f(x)$

$$M(\tilde{x}) = E(\tilde{x}) D(\tilde{x}) E(\tilde{x})^{-1}$$

$$E(\tilde{x}) = \left[n(\tilde{x}) \quad e_1(\tilde{x}) \quad ..... \quad e_{d-1}(\tilde{x}) \right]$$

$$D(\tilde{x}) = diag\left[\lambda_1(\tilde{x}) .... \lambda_d(\tilde{x}) \right]$$

Stretching/compression to guide flow



Conditions

- Compression in normal direction $\lambda_1(\tilde{x}) \leq 1,$
- Stretching in tangential direction $\lambda_i(\tilde{x}) \geq 1,$

  $i = 1 \dots d-1$

$$\begin{cases} \lambda_1(\tilde{x}) = 1 - \dfrac{1}{\Gamma(\tilde{x})} \\ \lambda_i(\tilde{x}) = 1 + \dfrac{1}{\Gamma(\tilde{x})} \end{cases}$$

- No effect far away $i = 1 \dots d-1$ $\quad \lim_{\Gamma(\tilde{x}) \to \infty} \lambda_i(\tilde{x}) = 1$

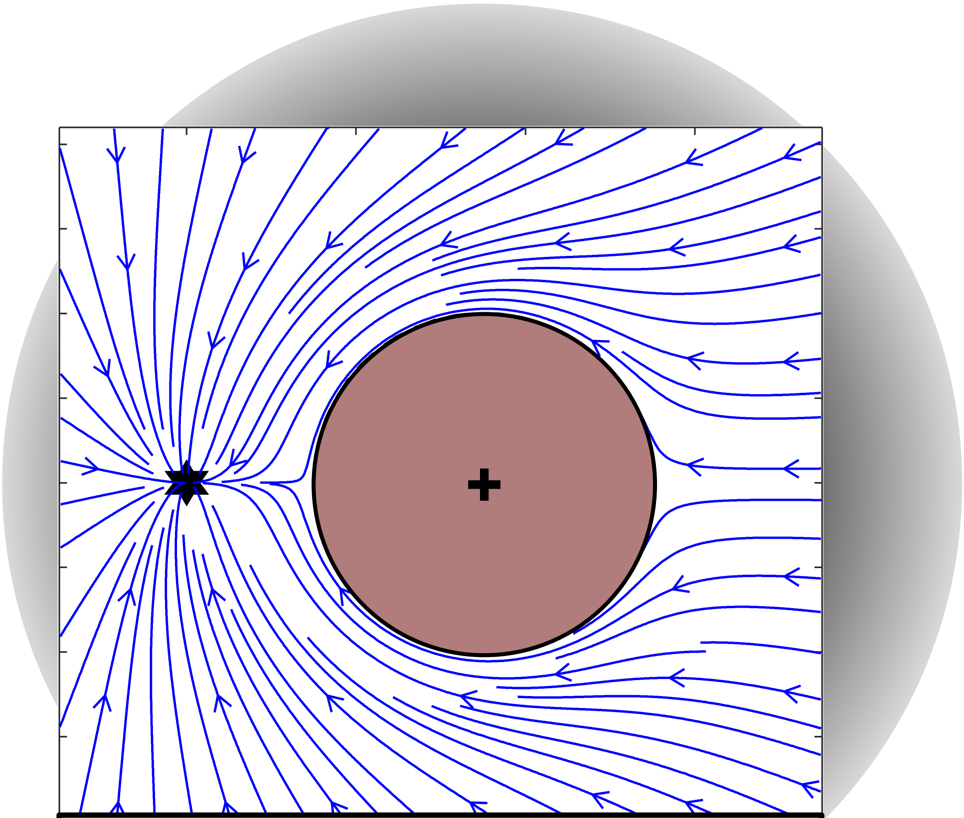
[1] Khansari-Zadeh, S. M., & Billard, A. (2012). A dynamical system approach to realtime obstacle avoidance. *Autonomous Robots*, *32*(4), 433-454.

Properties:

The attractor remains a stable point of the system.

The flow is ensured to never penetrate the obstacle.

See book for proofs.



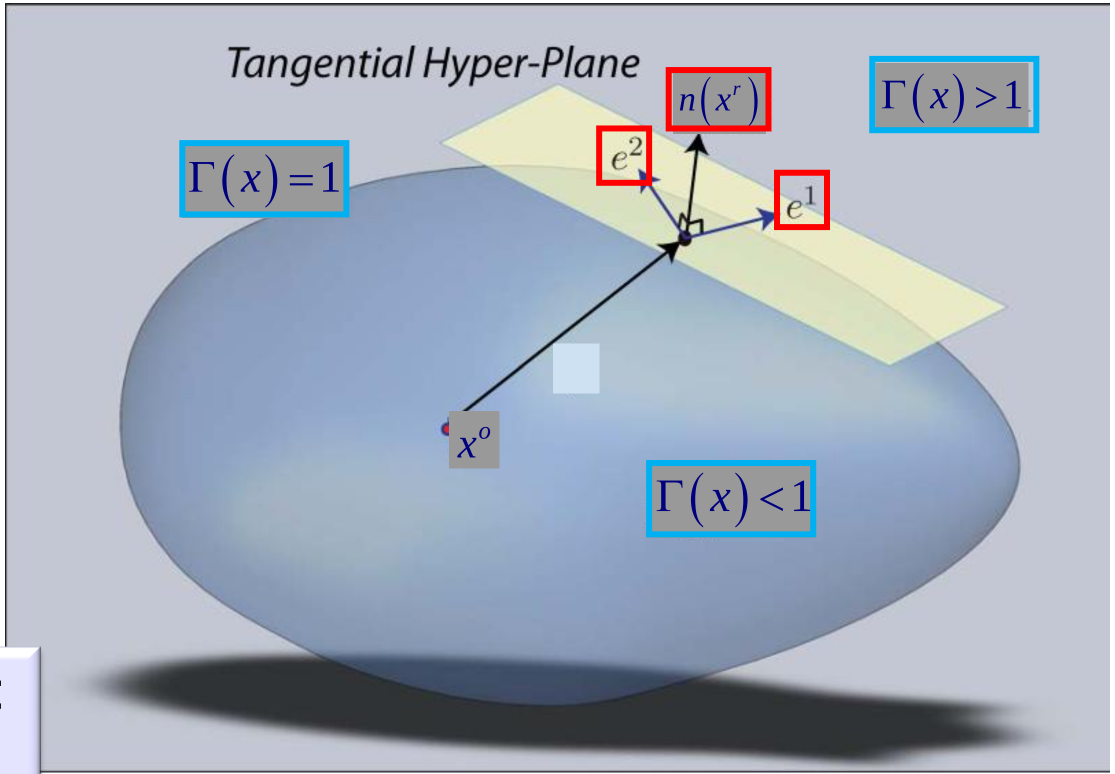
Modulated dynamics

3D Deflection - Representation

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

$$E(\tilde{x}) = \left[\boxed{n(\tilde{x})}\ \boxed{e_1(\tilde{x})}\ e_1(\tilde{x})_{d-1}\right]$$

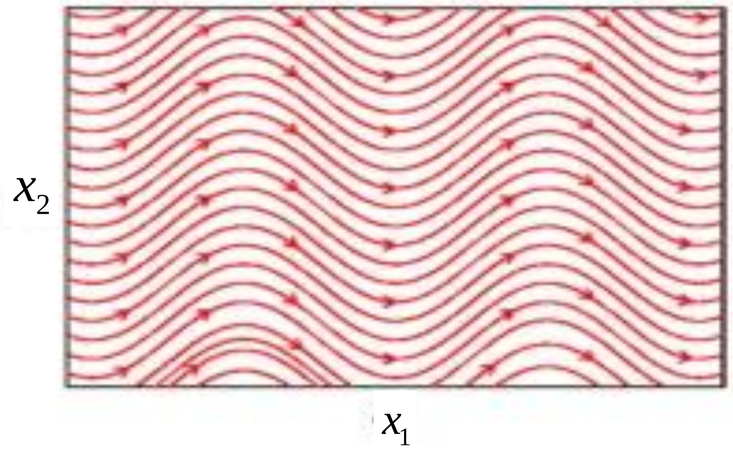
$$D(\tilde{x}) = diag\left[\lambda_1(\tilde{x})....\lambda_d(\tilde{x})\right]$$



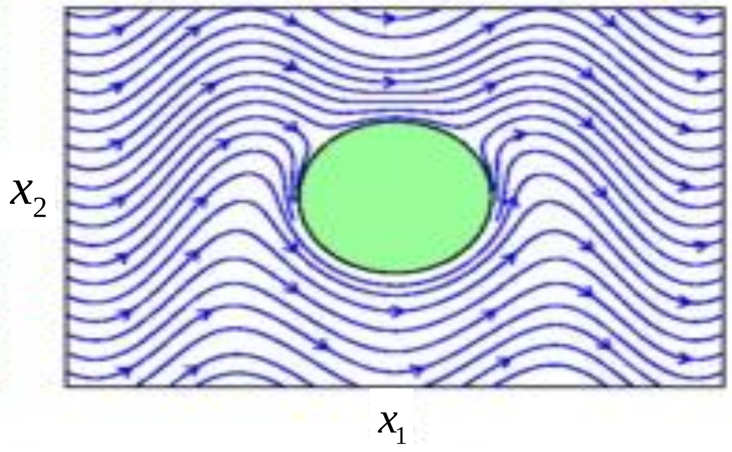
Distance to the obstacle:

$$\Gamma(x) : \mathbb{R}^N \to \mathbb{R}$$

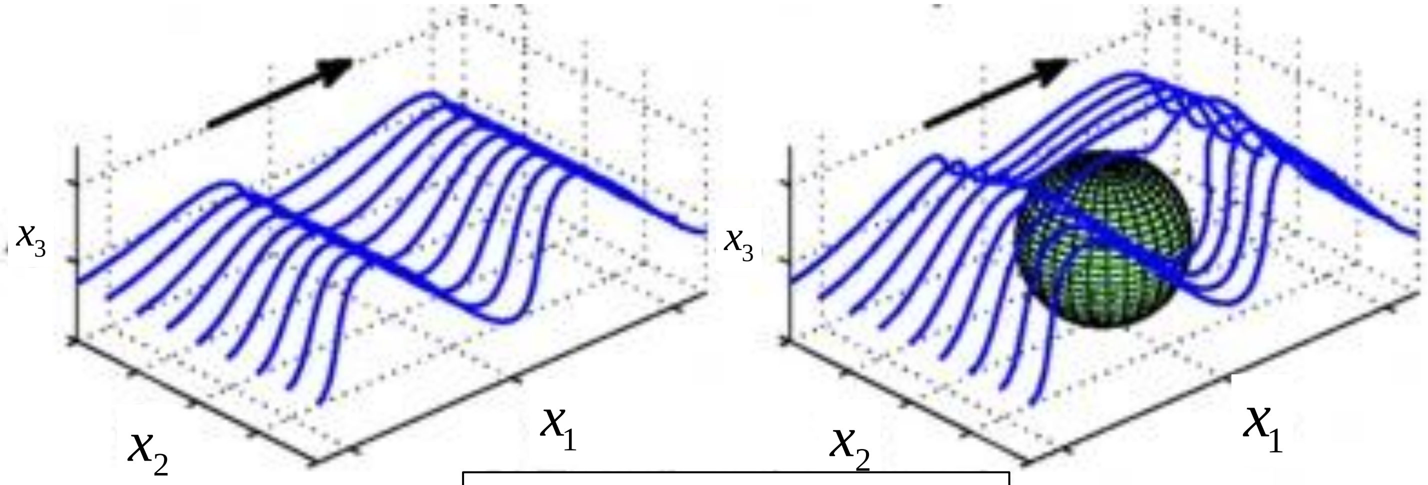
Modulation applies also to flows without attractors and to higher dimensions



Nominal DS: $\dot{x} = \sin(x)$



Modulated DS: $\dot{x} = M(x)\sin(x)$



3D flow

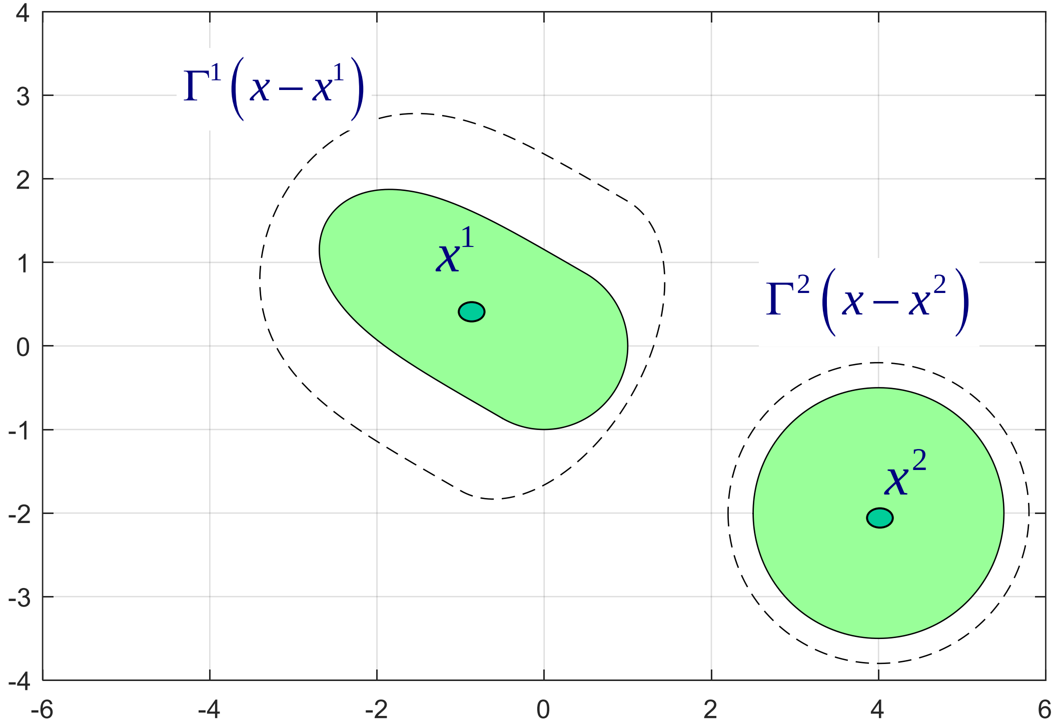
Extension to multiple obstacles

For each obstacle, define:

x^i : center of each obstacle

$\Gamma^i\left(x - x^i\right)$: distance function

$M_i\left(x - x^i\right)$: Modulation function



See today's and next week's exercises on how to compute distance functions depending on shape of obstacle.

Extension to multiple obstacles

For each obstacle, define:

x^i : center of each obstacle

$\Gamma^i\left(x-x^i\right)$: distance function

$M_i\left(x-x^i\right)$: Modulation function

Final modulation - product of obstacles' modulations:

$$\dot{x} = \prod_{i=1}^{nb \text{ obstacles}} M_i\left(x-x^i\right) f\left(x\right)$$

Modulate each set of eigenvalues

$$\lambda_1^i\left(x-x^i\right) = 1 - \frac{w^i\left(x-x^i\right)}{\Gamma^i\left(x-x^i\right)}$$

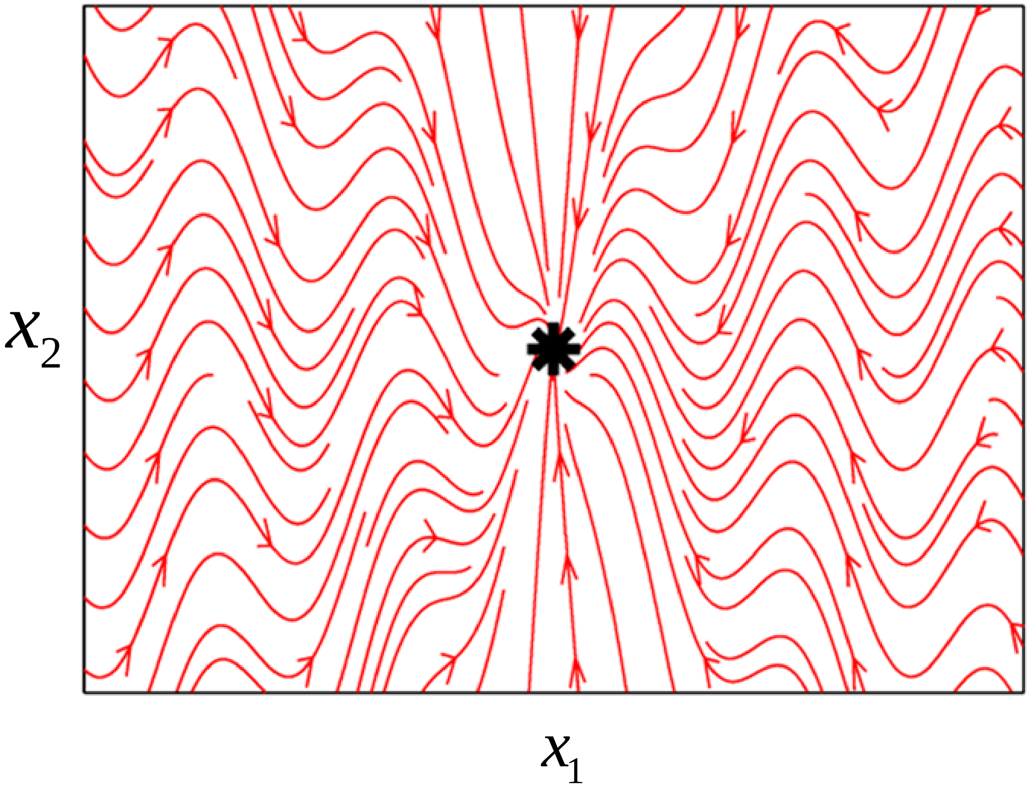
$$\lambda_d^i\left(x-x^i\right) = 1 + \frac{w^i\left(x-x^i\right)}{\Gamma^i\left(x-x^i\right)}, \quad d = 2 \ldots D$$

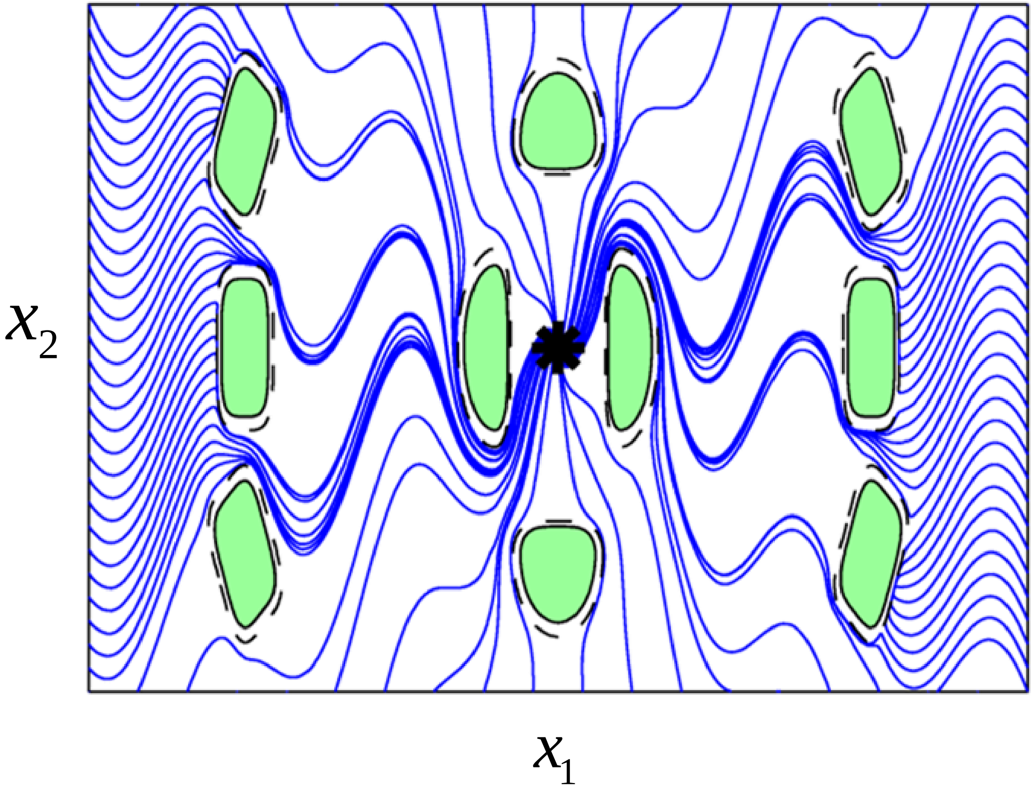
$$w^i\left(x-x^i\right) = \prod_{k=1, i\neq k}^{nb \text{ obstacles}} \frac{\left(\Gamma^k\left(x-x^k\right)-1\right)}{\left(\Gamma^k\left(x-x^k\right)-1\right)+\left(\Gamma^i\left(x-x^i\right)-1\right)}$$: balances relative effect of each obstacle

Extension to multiple obstacles

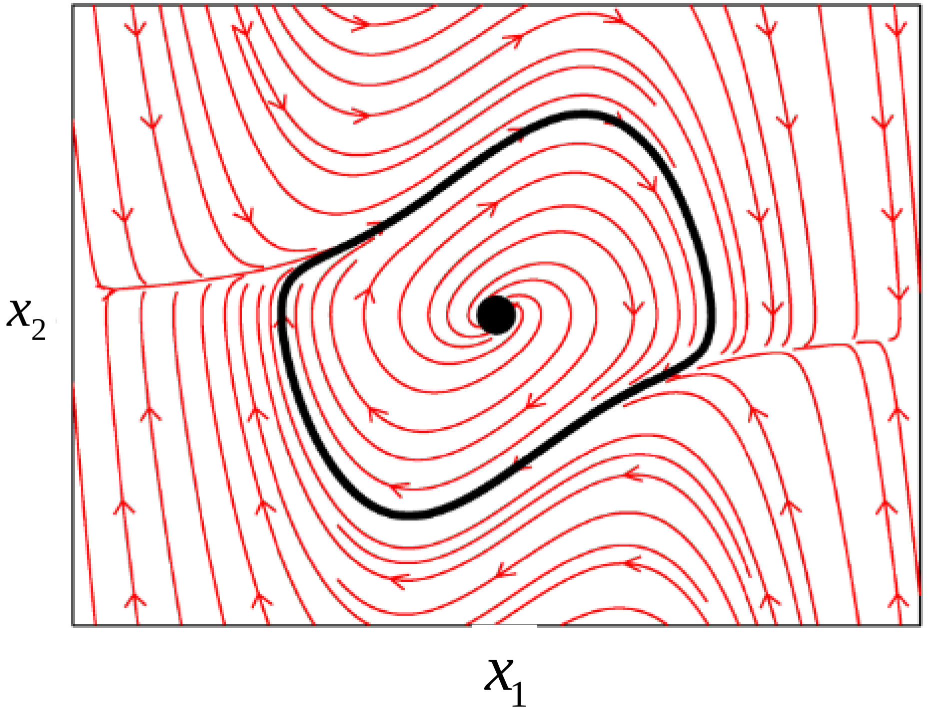
Final modulation - product of obstacles' modulations:

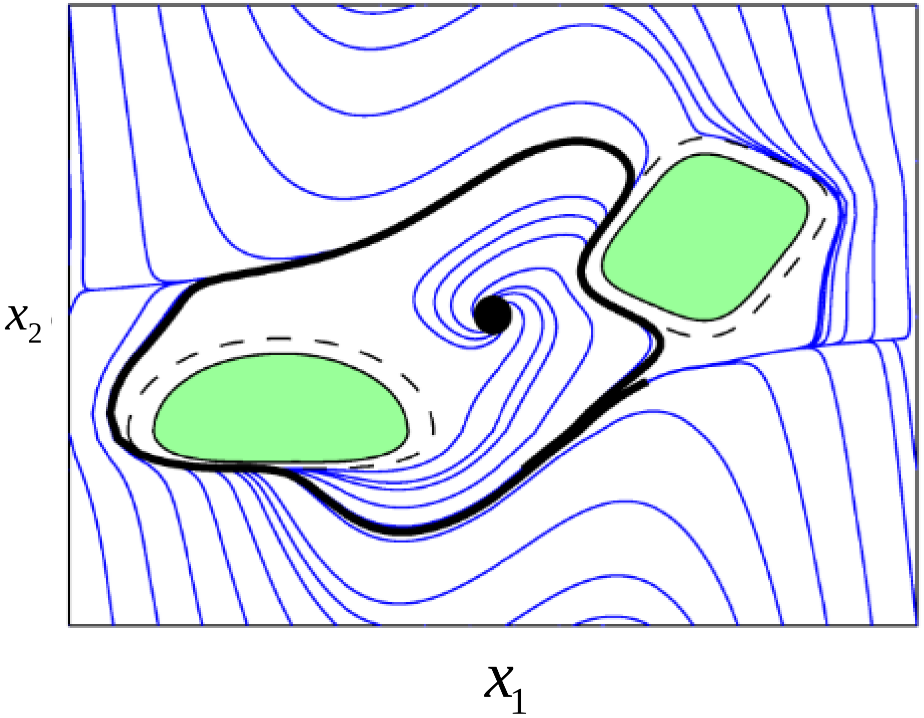
$$\dot{x} = \prod_{i=1}^{nb\ \text{obstacles}} M_i\left(x - x^i\right) f\left(x\right)$$





Modulation applies also to flows without attractors





Nominal DS - $\dot{x} = f(x)$

nonlinear DS with limit cycle

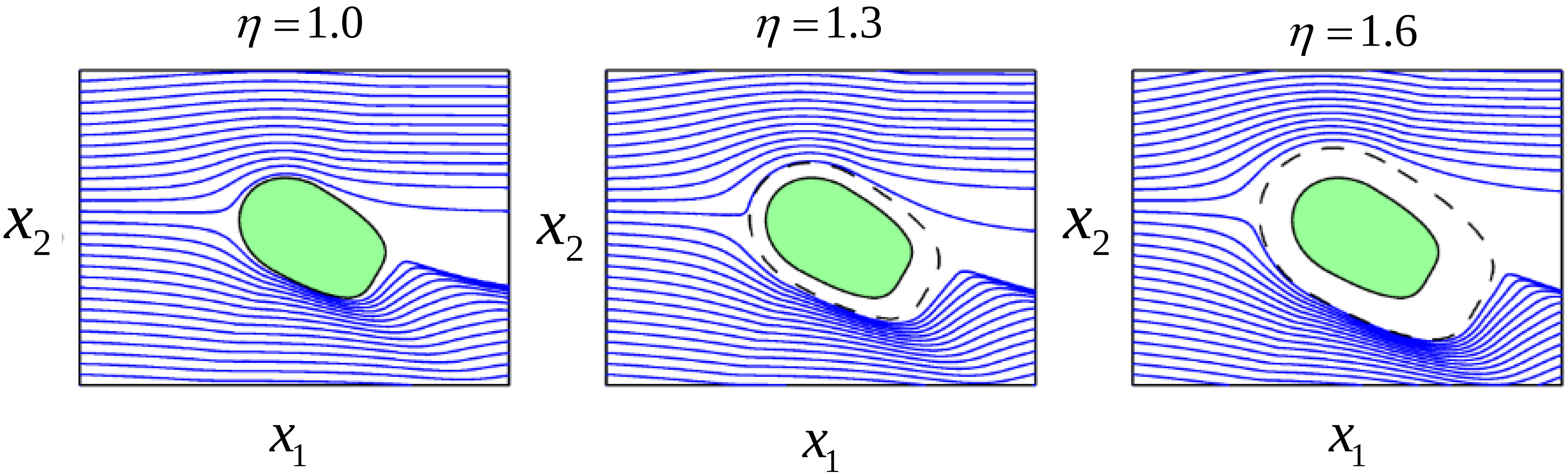
$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -x_1 + 0.9x_2\left(1-(x_1)^2\right) \end{cases}$$

Modulated DS: $\dot{x} = \prod_{i=1}^{2} M_i\left(x - x^i\right) f(x)$.

Safety Factor

When the obstacle is fragile or dangerous, it may be necessary to define a **safety margin** around the obstacle. Such margin can be obtained through a scaling of the state variable.

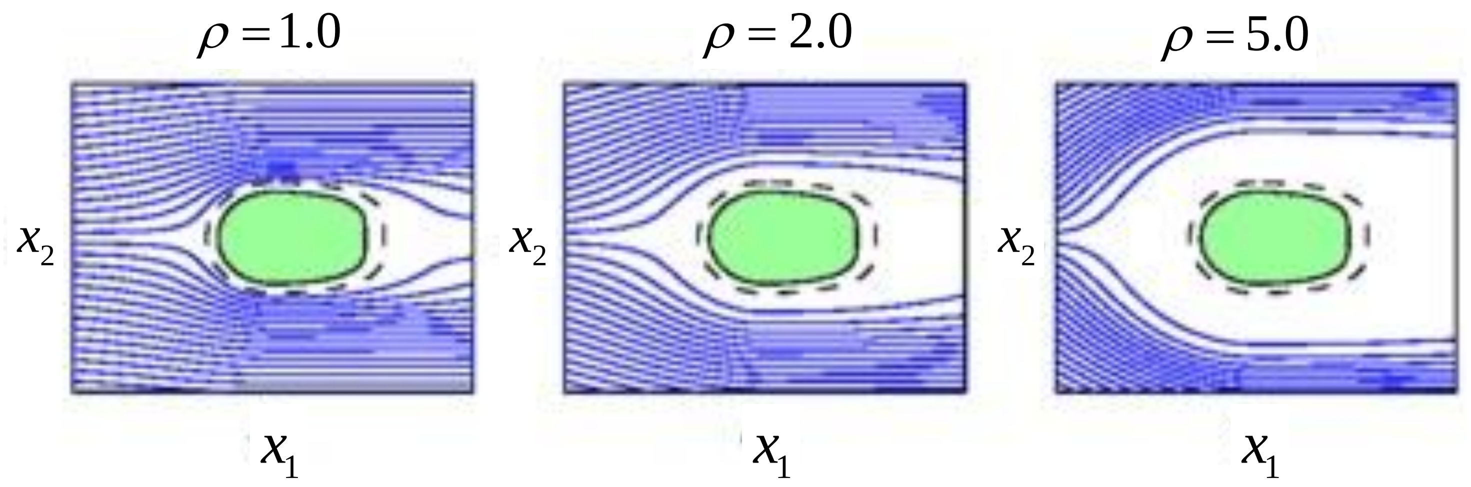
$$M\left(\frac{1}{\eta}\tilde{x}\right) = E\left(\frac{1}{\eta}\tilde{x}\right)\Lambda\left(\frac{1}{\eta}\tilde{x}\right)E\left(\frac{1}{\eta}\tilde{x}\right), \quad \text{scaling: } \eta \geq 1.$$



Reactivity

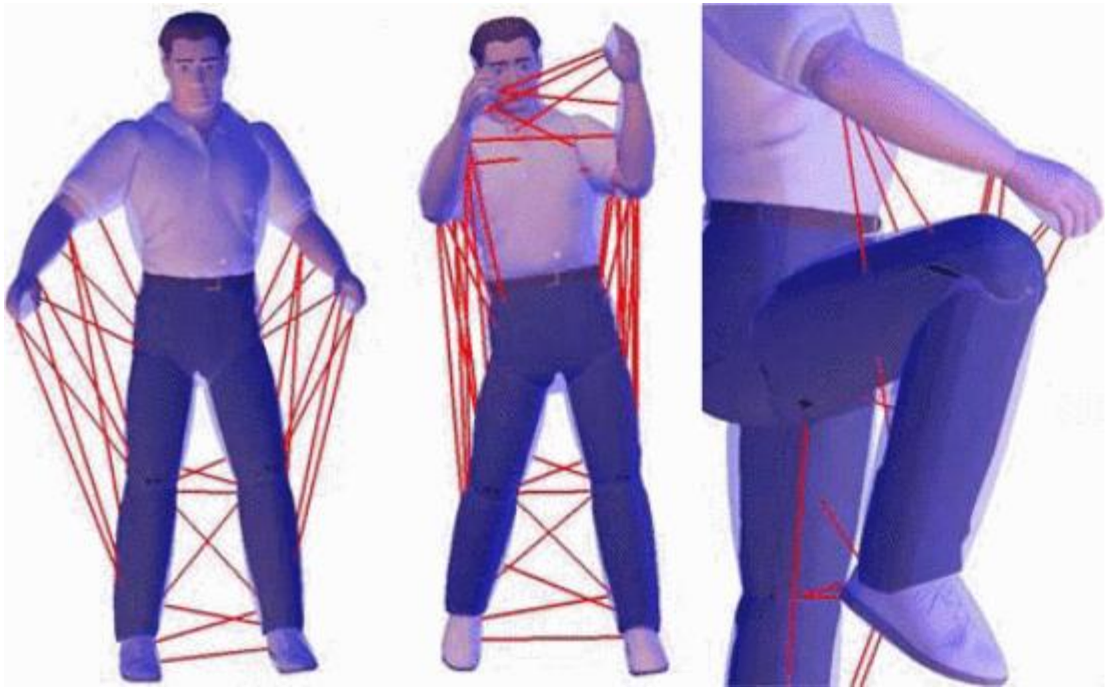
The **magnitude of the modulation** created by the obstacle can be tuned by **modifying the eigenvalues** of the dynamic modulation matrix through a ***reactivity factor***. *The larger the reactivity, the larger the amplitude of the deflection*, and consequently the earlier the robot responds to the presence of an obstacle.

$$\begin{cases} \lambda_1(\tilde{x}) = 1 - \dfrac{1}{\left|\Gamma(\tilde{x})\right|^{\frac{1}{\rho}}} \\ \lambda_d(\tilde{x}) = 1 + \dfrac{1}{\left|\Gamma(\tilde{x})\right|^{\frac{1}{\rho}}} \end{cases}, \quad \text{reactivity factor: } \rho > 0.$$



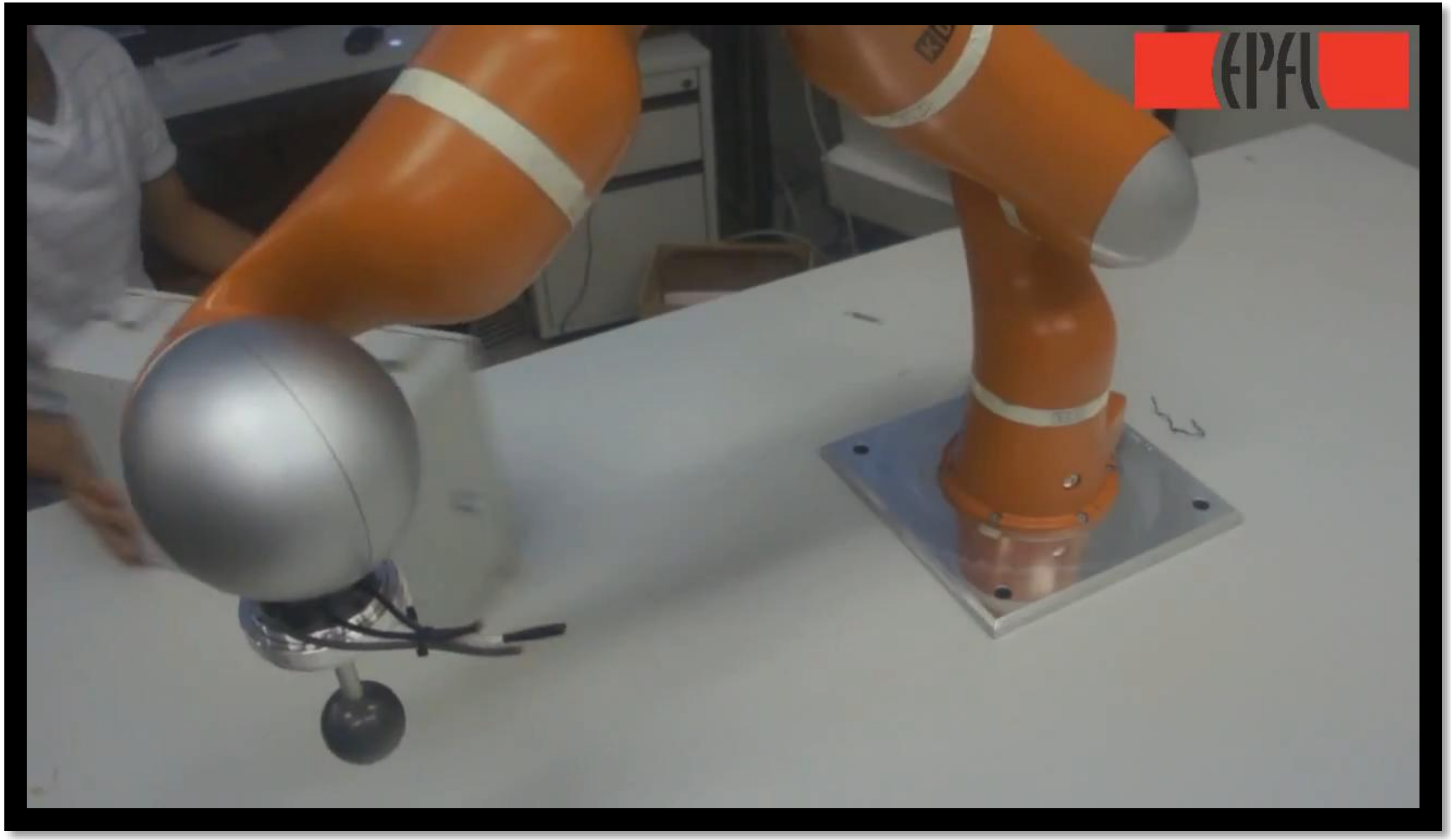
Convexifying objects

The modulation presented in the previous slides *requires that the object be convex*. One can build a convex hull around non-convex obstacles, see Benallegue et al. for a fast method.



Benallegue, M., Escande, A., Miossec, S., & Kheddar, A. (2009). Fast C1 proximity queries using support mapping of sphere-torus-patches bounding volumes. In *Proc. IEEE int. conf. on robotics and automation* (pp. 483–488).

Robotics demonstration: live obstacle avoidance of convex obstacle



[1] Khansari-Zadeh, S. M., & Billard, A. (2012). A dynamical system approach to realtime obstacle avoidance. *Autonomous Robots*, *32*(4), 433-454.

Formal Guarantees

M(x) guarantees that a flow starting outside the obstacle will never penetrate the obstacle.

$$\forall x(t_0),\ \ \Gamma(\tilde{x}) \geq 1,\ \text{with dynamics}\ \dot{x} = M(\tilde{x})\, f(x),\ \ \Gamma(\tilde{x}(t)) \geq 1,\ \ t \geq t.$$

M(x) guarantees that the attractor remains a stable point, if the attractor is outside the obstacle.

$$\text{If } x^*,\ \ \Gamma(\tilde{x}^*) \geq 1,\ \text{with dynamics}\ \dot{x} = f(x^*) = 0,\ \ M(\tilde{x}^*)\, f(x^*) = 0.$$

The attractor is, however, no longer the unique fixed point of the system!

Spurious Fixed Points

The attractor is no longer the unique fixed point of the system.

Uniqueness of the attractor: $\Rightarrow M(\tilde{x}) f(x) \neq 0, \ \forall x \neq x^*.$

$$M(\tilde{x}) = E(\tilde{x}) D(\tilde{x}) E(\tilde{x})^{-1} \qquad E(\tilde{x}) = \left[n(\tilde{x}) \quad e_1(\tilde{x}) \quad e_1(\tilde{x})_{d-1} \right]$$

$$\Lambda(\tilde{x}) = diag\left[\lambda_1(\tilde{x}) .... \lambda_d(\tilde{x}) \right]$$

Vanishes at obstacle's boundary

All positives

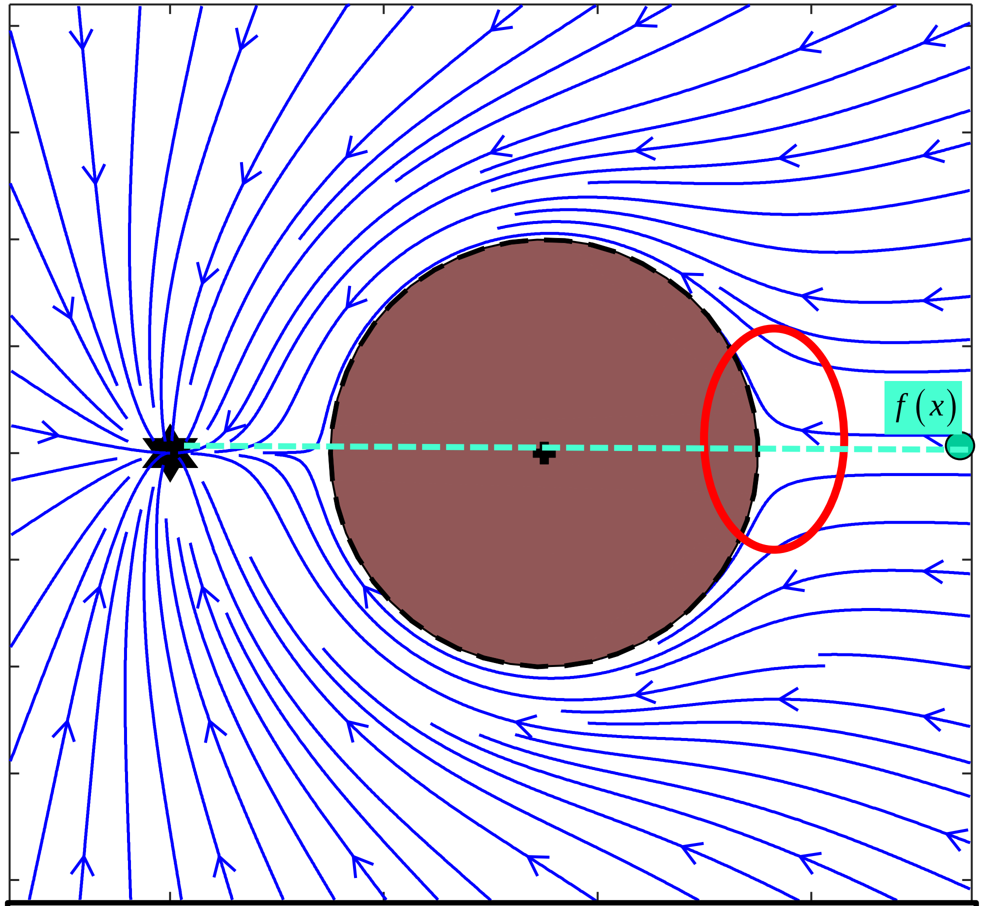
As long as f(x) has a tangential component, it will not vanish.

Fixed points: All points x such that $n(x)^T \frac{f(x)}{\|f(x)\|} = \pm 1, \ \Gamma(x) = 1,$ become fixed points.

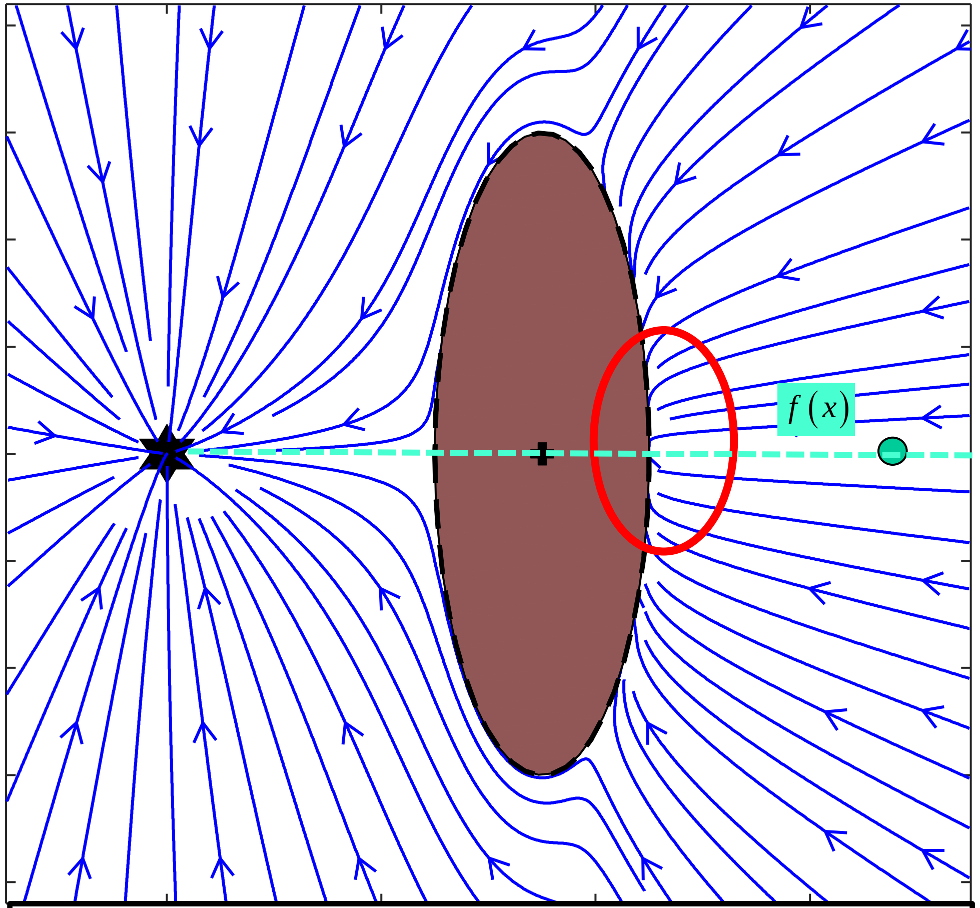
The dynamics vanishes for points on the obstacle's boundary and for which the boundary's normal is colinear to f.

EPFL

Spurious Fixed Points



Circular obstacle with convergence for all but one trajectory.



Cylindrical obstacle modulating a linear DS resulting in one spurious attractor.

Limitation of orthonormal basis for the deflection

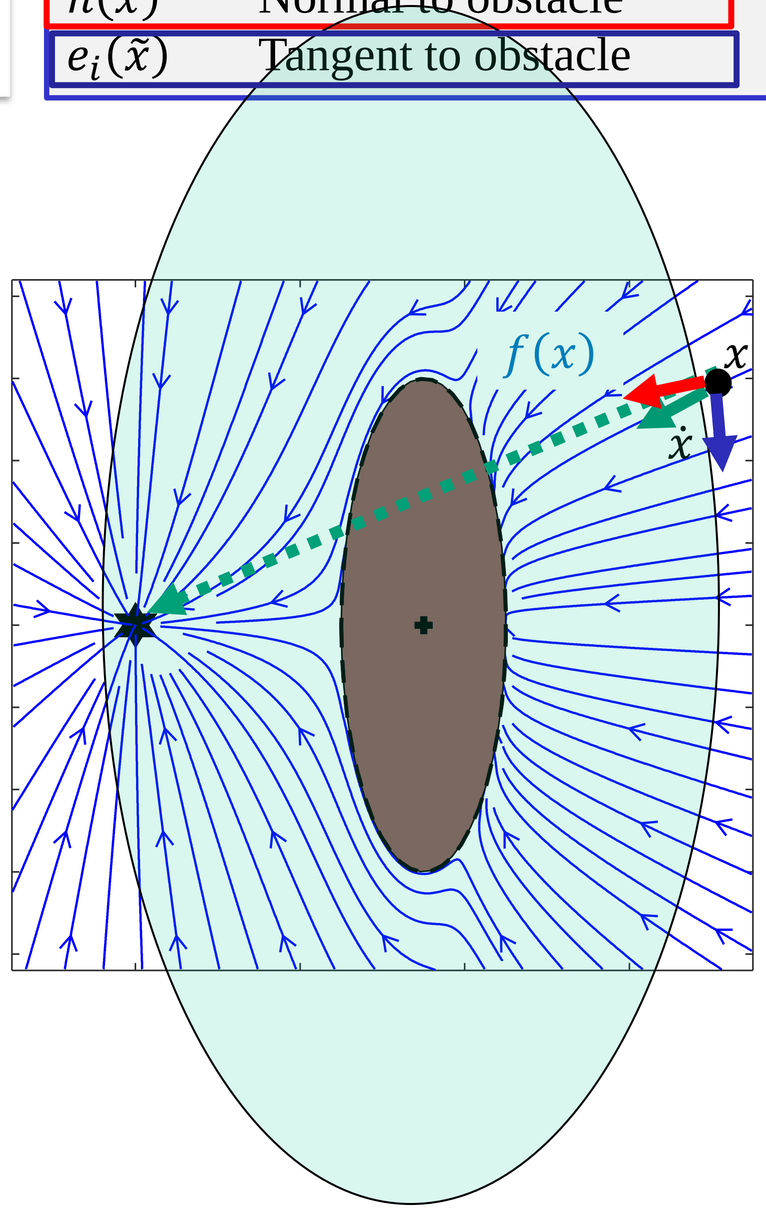
Legend

$\tilde{x}$	Robot's relative state ($\tilde{x} = x - x^o$)
$E(\tilde{x})$	Decomposition matrix
$n(\tilde{x})$	Normal to obstacle
$e_i(\tilde{x})$	Tangent to obstacle

Modulation is given by: $\dot{x} = M(\tilde{x})\, f(x)$

$$M(\tilde{x}) = E(\tilde{x}) D(\tilde{x}) E(\tilde{x})^{-1}$$

$$E(\tilde{x}) = \left[n(\tilde{x}) \quad e_1(\tilde{x}) \quad \ldots\ldots \quad e_1(\tilde{x})_{d-1} \right]$$



Limitation of orthonormal basis for the deflection

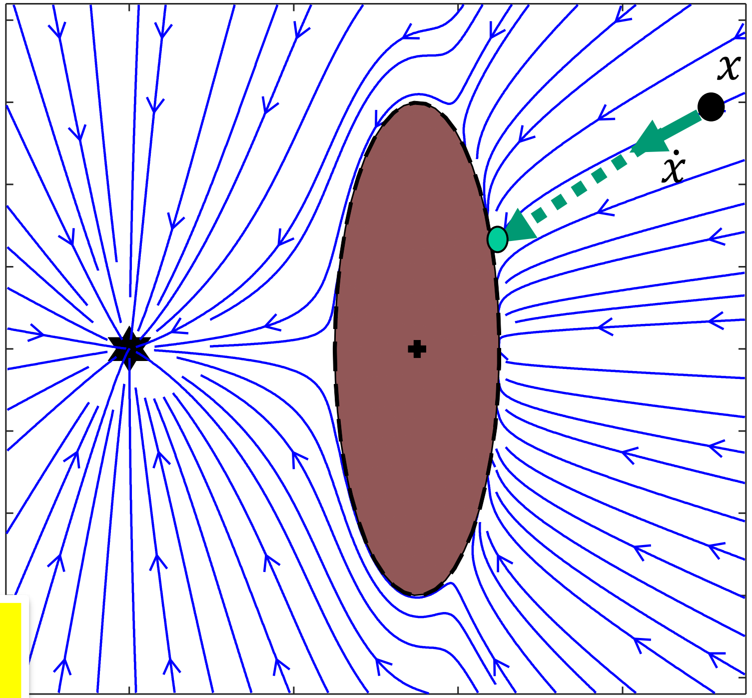
Legend

$\tilde{x}$	Robot's relative state ($\tilde{x} = x - x^o$)
$E(\tilde{x})$	Decomposition matrix
$n(\tilde{x})$	Normal to obstacle
$e_i(\tilde{x})$	Tangent to obstacle

Modulation is given by: $\dot{x} = M(\tilde{x})\, f(x)$

$$M(\tilde{x}) = E(\tilde{x}) D(\tilde{x}) E(\tilde{x})^{-1}$$

$$E(\tilde{x}) = \left[n(\tilde{x}) \quad e_1(\tilde{x}) \quad \ldots\ldots \quad e_1(\tilde{x})_{d-1} \right]$$



Multiple paths now converge to the spurious attractor.

Idea: Relax the constraint of an orthonormal basis.
It is sufficient for the vector to be **linearly independent** for E to be invertible.

Lukas Huber, Aude Billard, and Jean-Jacques Slotine. *IEEE Robotics and Automation Letters* 4.2 (2019): 1462-1469.

Legend	
$\tilde{x} \in \mathbb{R}^d$	Robot's relative state ($\tilde{x} = x - x^o$)
$E(\tilde{x})$	Decomposition matrix
$r(\tilde{x})$	Reference direction

Non-orthonormal basis for the deflection

Modulation is given by: $\dot{x} = M(\tilde{x}) f(x)$

$$M(\tilde{x}) = E(\tilde{x}) D(\tilde{x}) E(\tilde{x})^{-1}$$

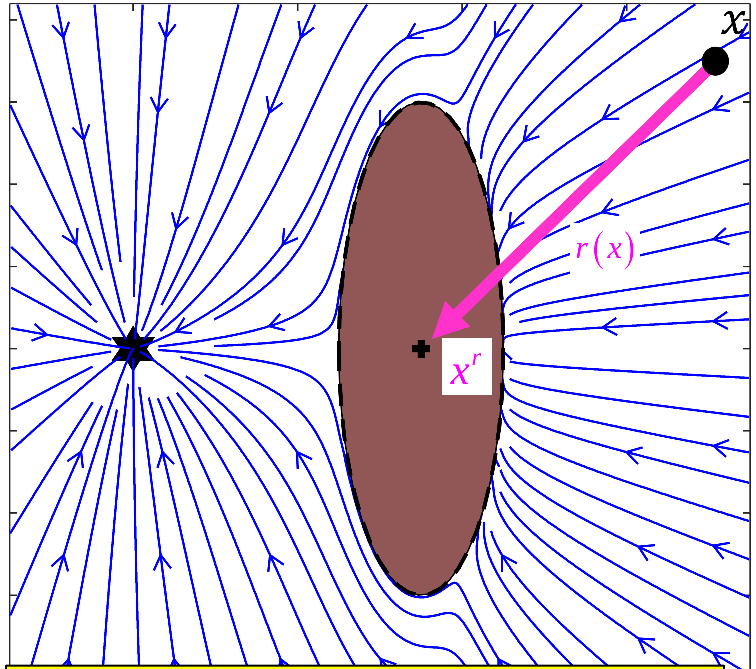
$$E(\tilde{x}) = \left[r(\tilde{x}) \quad e_1(\tilde{x}) \quad \ldots\ldots \quad e_1(\tilde{x})_{d-1} \right]$$

Let r be an arbitrary vector that is linearly independent of all e_i.

Define a reference point x^r inside the obstacle as new center of the modulation and set:

$$r(\tilde{x}) = \frac{x - x^r}{\|x - x^r\|} = \frac{\tilde{x}}{\|\tilde{x}\|}.$$

The new decomposition matrix $E(\tilde{x})$ is not orthogonal anymore, but is full rank.



If x^r is inside the obstacle, $r(x)$ is linearly independent of all tangents e to the surface.

Non-orthonormal basis for the deflection

Legend

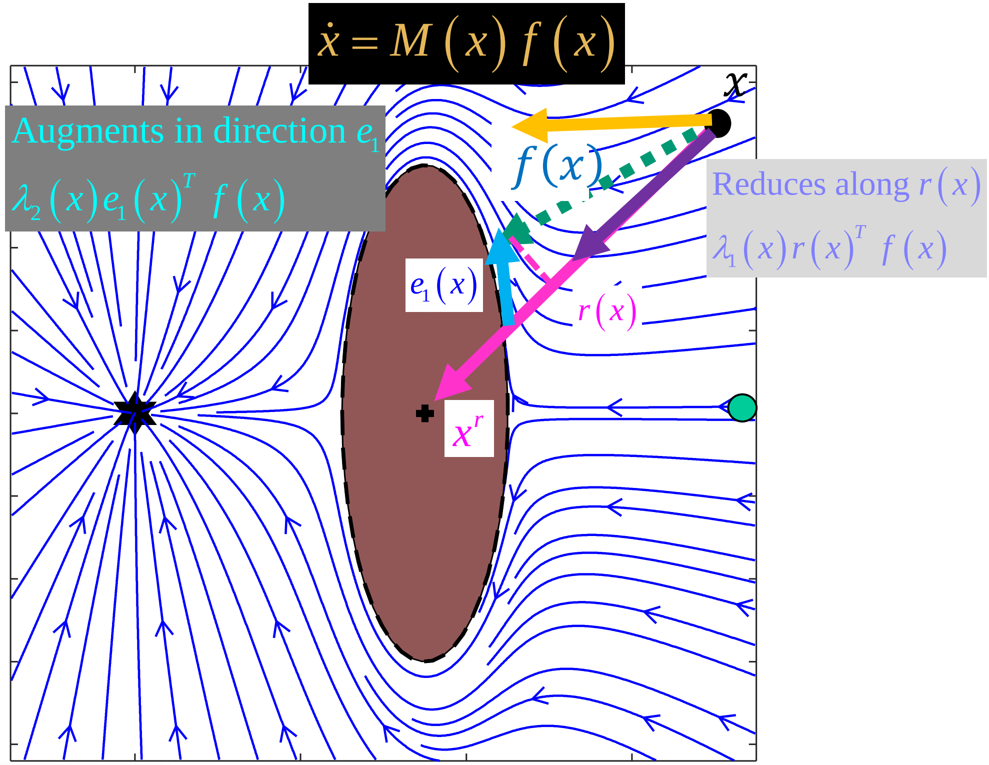
$\tilde{x} \in \mathbb{R}^d$	Robot's relative state ($\tilde{x} = x - x^o$)
$E(\tilde{x})$	Decomposition matrix
$r(\tilde{x})$	Reference direction

Modulation is given by: $\dot{x} = M(\tilde{x}) f(x)$

$$M(\tilde{x}) = E(\tilde{x}) D(\tilde{x}) E(\tilde{x})^{-1}$$

$$E(\tilde{x}) = \left[\underline{r(\tilde{x})} \quad e_1(\tilde{x}) \quad \ldots.. \quad e_1(\tilde{x})_{d-1} \right]$$

$$\begin{cases} \lambda_1(\tilde{x}) = 1 - \dfrac{1}{\Gamma(\tilde{x})} \\ \lambda_2(\tilde{x}) = 1 + \dfrac{1}{\Gamma(\tilde{x})} \end{cases}$$



The flow is deflected away from the spurious fixed point.

It remains only a single infinitesimally small trajectory converging to the spurious fixed point.

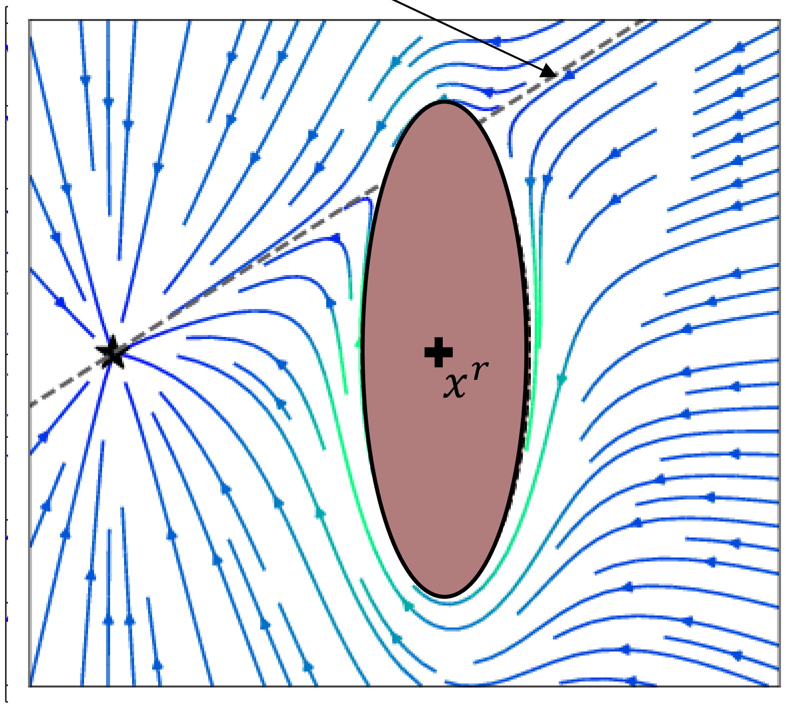
This is a *saddle* point, as any small disturbance would send the flow away from the point.

Effect of Choice of Reference Direction

The choice of reference direction influences the way the dynamics avoids the obstacle.

Still only one single trajectory to a single saddle point.

$$r(\tilde{x}) = \frac{x - x^r}{\left\| x - x^r \right\|} = \frac{\tilde{x}}{\left\| \tilde{x} \right\|}$$



We will see that this reference point can allow us to avoid concave obstacles and preserve convergence to the original attractor, see 2nd part.

SUMMARY – 1st Part

- The DS modulation approach has been extended to enable obstacle avoidance **in closed form.**
  → This ensures fast and highly reactive obstacle avoidance

- It preserves guarantee of **asymptotic convergence to the goal**.
  → This ensures that the robot returns to its path once the obstacle has been avoided without the need for re-planning the trajectory.

- The approach is **parameterized by two hyperparameters** that determine the **safety margin** and the **reactivity**.
  → These are intuitive parameters that can be easily fixed by hand knowing the geometry of the obstacle.

Limitations

- The obstacle's boundary must be convex.
- Spurious attractors may arise on the obstacle's boundary.

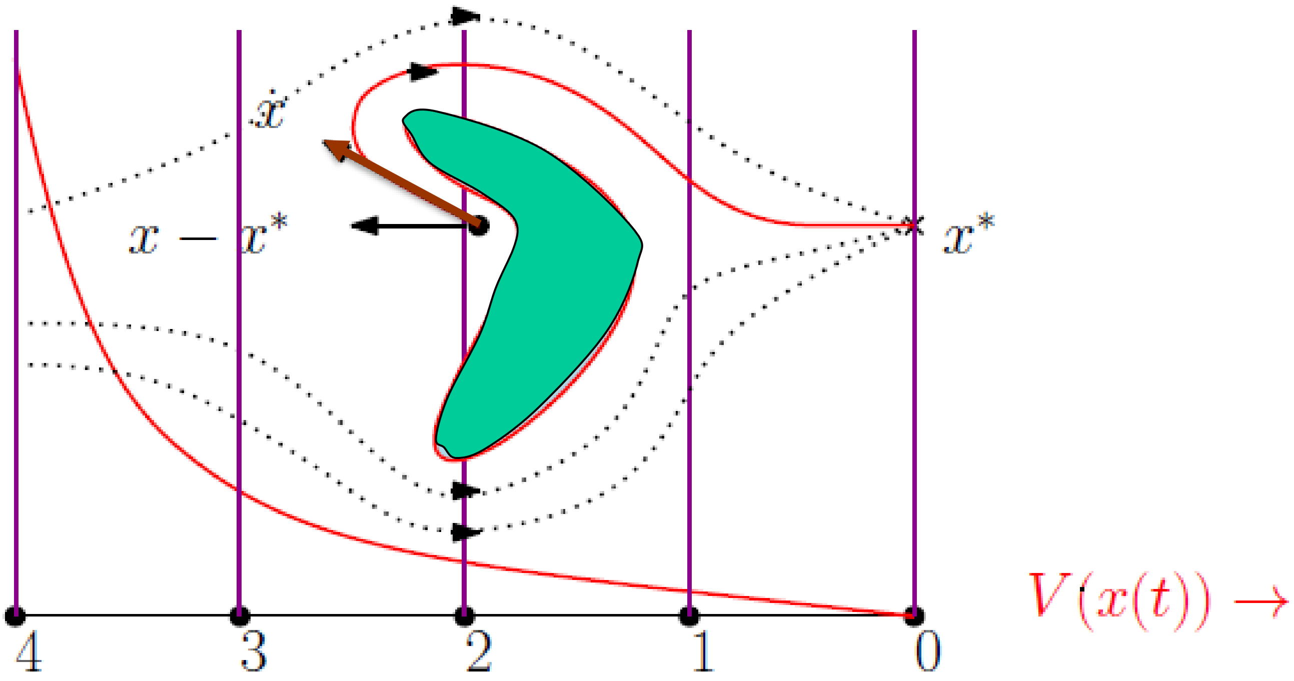
DS-based Obstacle Avoidance of Concave Objects

- Extends the modulation to enable obstacle avoidance for a class of concave obstacles
- Offers convergence guarantees using Contraction Theory.
- Reduces the number of spurious attractors to a unique trajectory, easily avoidable.
- Invert the model to enclose the robot into a fixed volume, from which the robot cannot escape.

- Lukas Huber, Aude Billard, and Jean-Jacques Slotine. "Avoidance of Convex and Concave Obstacles With Convergence Ensured Through Contraction." *IEEE Robotics and Automation Letters* 4.2 (2019): 1462-1469.
- Lukas Huber, Jean-Jacques Slotine, and Aude Billard. "Avoiding Dense and Dynamic Obstacles in Enclosed Spaces: Application to Moving in Crowds. IEEE Transactions on Robotics, 2022.

Stability Guarantees

A DS is Lyapunov stable at x^* if $\dot{V}(x) = \frac{\partial V(x)}{\partial x} f(x) < 0, \quad \forall x \neq x^*$

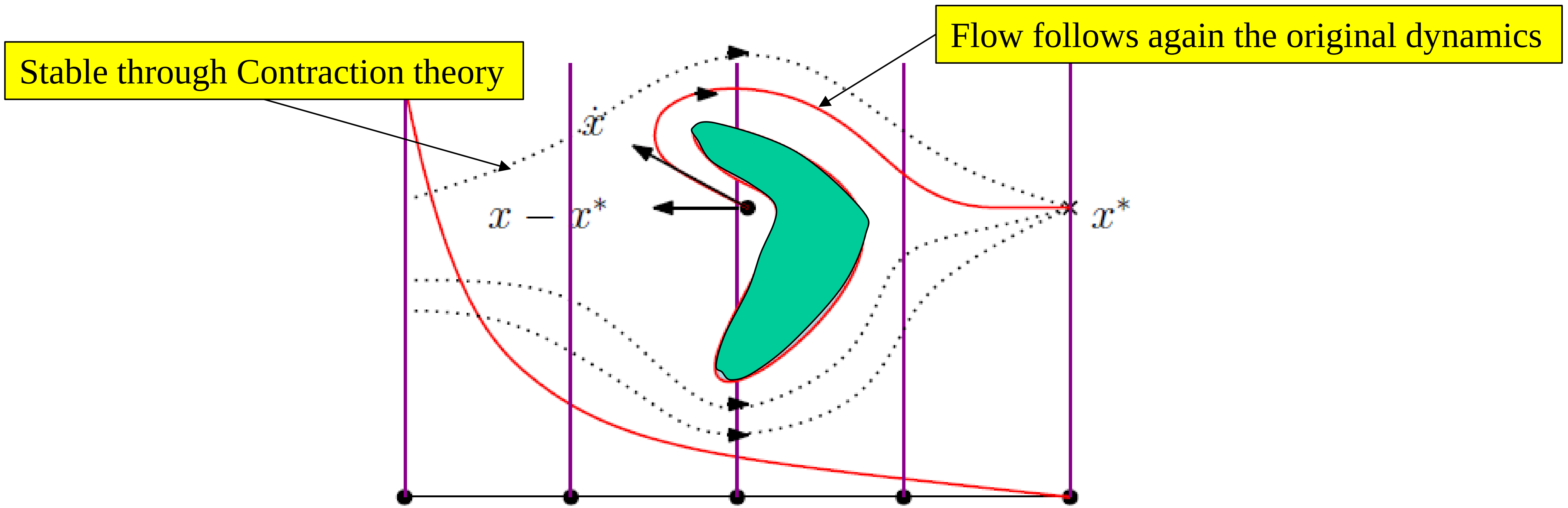


$V(x) = ||x||^2$

$\dot{x}$ violates the Lyapunov constraint, even for P-QLF.

Stability Guarantees

Idea: Use contraction theory to prove stability in the region where Lyapunov stability no longer applies.



We need a **metric** to guarantee that the system *contracts* and that the flow will eventually go back to the original DS.

Contraction: Stability of infinitesimals

Stability between finitely apart trajectories

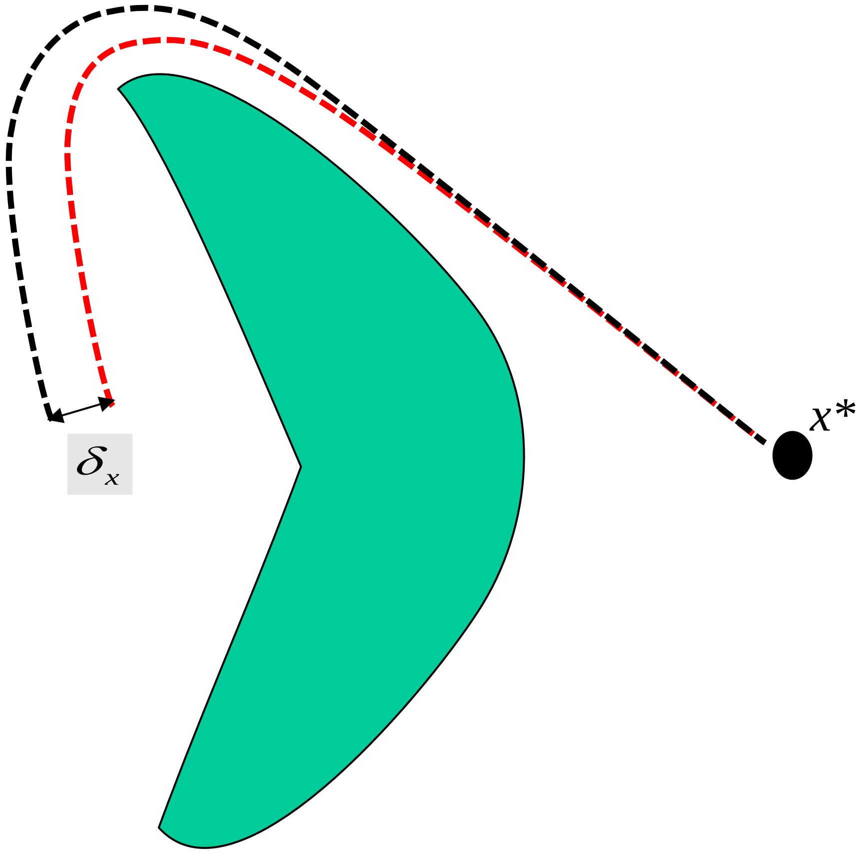
δ_x virtual displacement between two infinitesimally close trajectories

Define $V(x,\delta_x)=\delta_x^T \underbrace{P(x)}_{\succ 0}\delta_x$

$P(x)$: Contraction Metric

$V(x,\delta_x)$: Differential Control Lyapunov Function

Stability Condition: $\dot{V}(x,\delta_x)=-2\alpha V(x,\delta_x), \ \alpha>0$



If the above condition is satisfied, all the trajectories converge to one single trajectory incrementally and exponentially, regardless of the initial conditions.

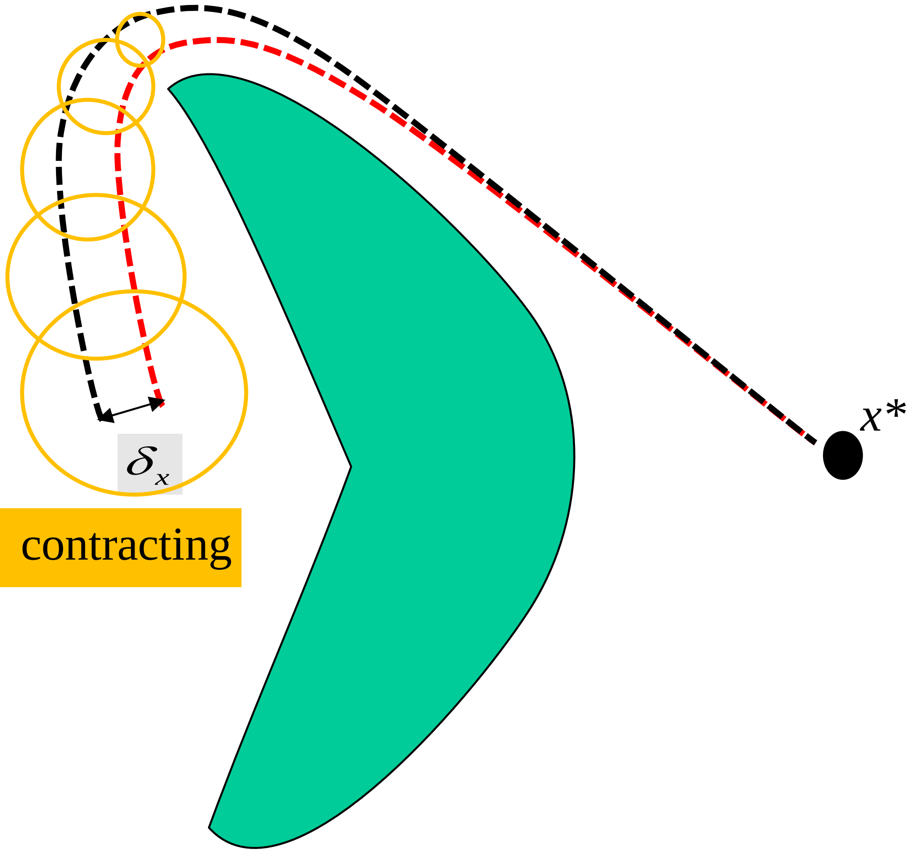
Contraction: Stability of infinitesimals

δ_x virtual displacement between two infinitesimally close trajectories

Define $V(x,\delta_x)=\delta_x^T \underbrace{P(x)}_{\succ 0}\delta_x$

$P(x)$: Contraction Metric

$V(x,\delta_x)$: Differential Control Lyapunov Function

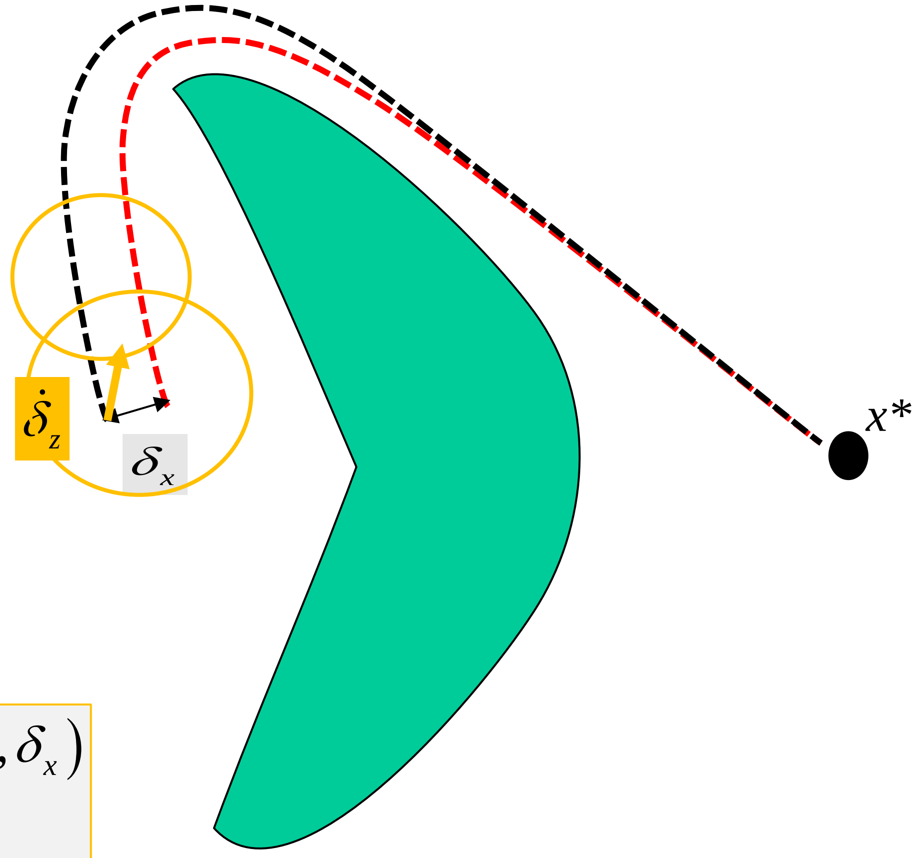


$P(x)$: contracting

Contraction: Stability of infinitesimals

δ_x virtual displacement between two infinitesimally close trajectories

$$\dot{x} = f(x) \Rightarrow \dot{\delta}_x = \frac{\partial f}{\partial x}(x)\delta_x$$



$$P(x) \succ 0$$
$$P(x) := \theta(x)^T \theta(x)$$
$$\delta_x^T P(x)\delta_x = \delta_x^T \theta(x)^T \theta(x)\delta_x$$
$$\delta_z := \theta(x)\delta_x$$
$$\Rightarrow \delta_x^T P(x)\delta_x = \delta_z^T \delta_z$$

$V(x,\delta_x) = \delta_x^T P(x)\delta_x$ and stability Condition: $\dot{V}(x,\delta_z) = -2\alpha V(x,\delta_x)$

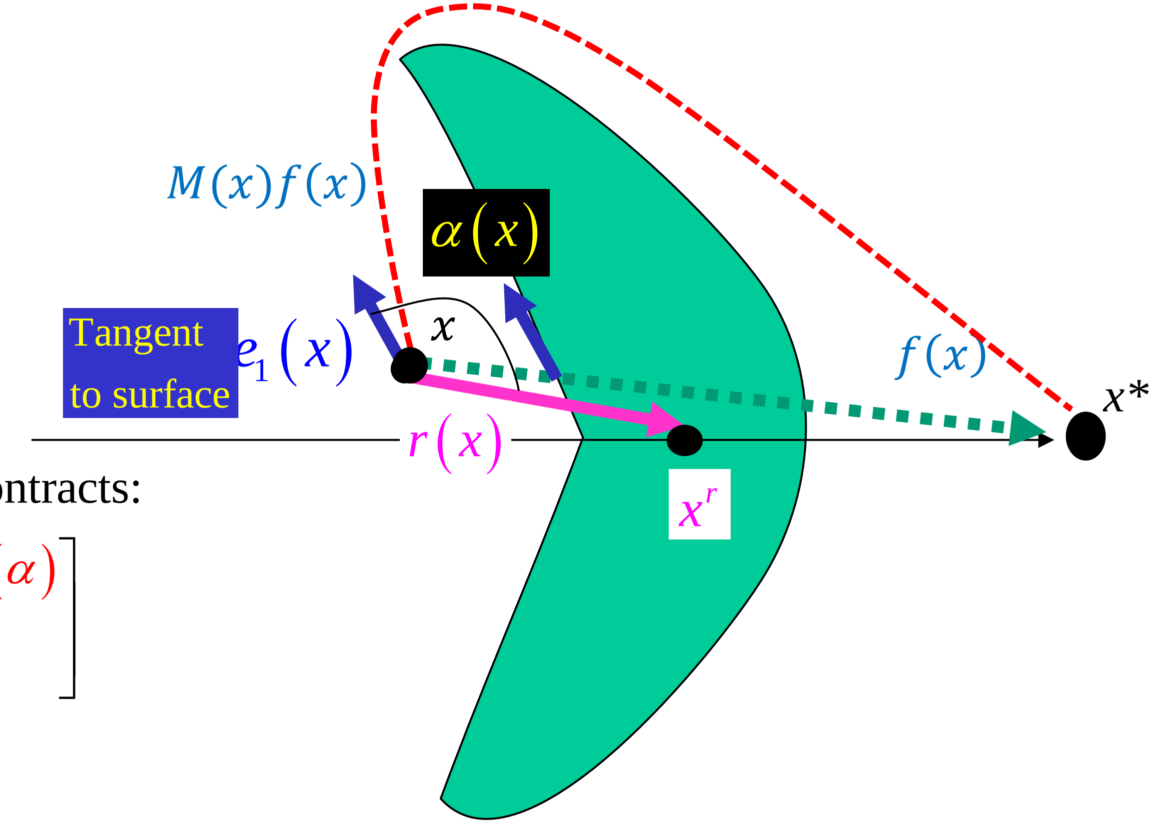
$$\Rightarrow \dot{V}(x,\delta_z) = -2\alpha\delta_x^T P(x)\delta_x = -2\alpha\delta_z^T\delta_z$$

$$\dot{P}(x) + P(x)\frac{\partial f(x)}{\partial x} + \frac{\partial f(x)}{\partial x}^T P(x) \le -2\alpha P(x)$$

New Metric

Define a new frame of reference:

$$\theta(x) = \begin{bmatrix} r(x) \\ e_1(x) \end{bmatrix} \qquad \delta_z := \theta(x)\delta_x$$

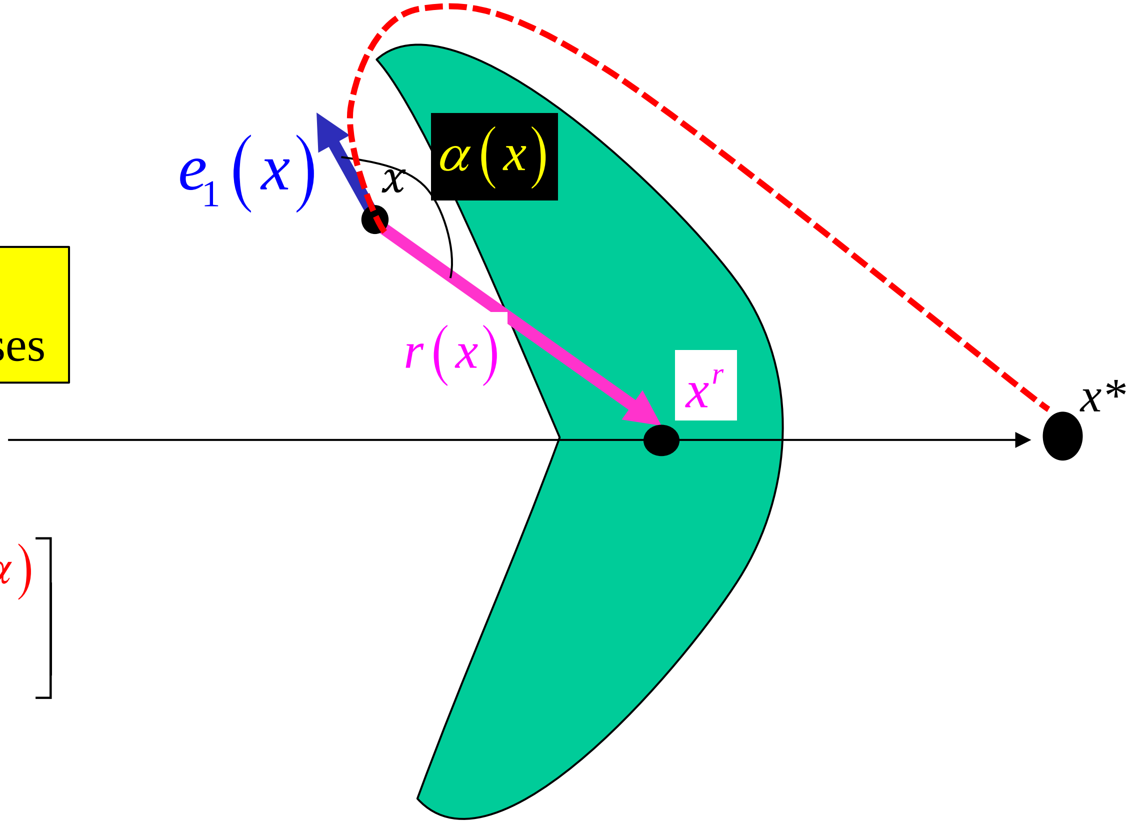


Define a state-dependent metric under which the system contracts:

$$P(x) = \theta(x)\theta(x)^T = \begin{bmatrix} r^2(x) & r(x)e_1(x)\cos(\alpha) \\ r(x)e_1(x)\cos(\alpha) & (e_1(x))^2 \end{bmatrix}$$

$$\boxed{\dot{P}(x) + P(x)\frac{\partial f(x)}{\partial x} + \frac{\partial f(x)}{\partial x}^T P(x) \leq -2\alpha P(x)}$$

Temporal Evolution of the Metric



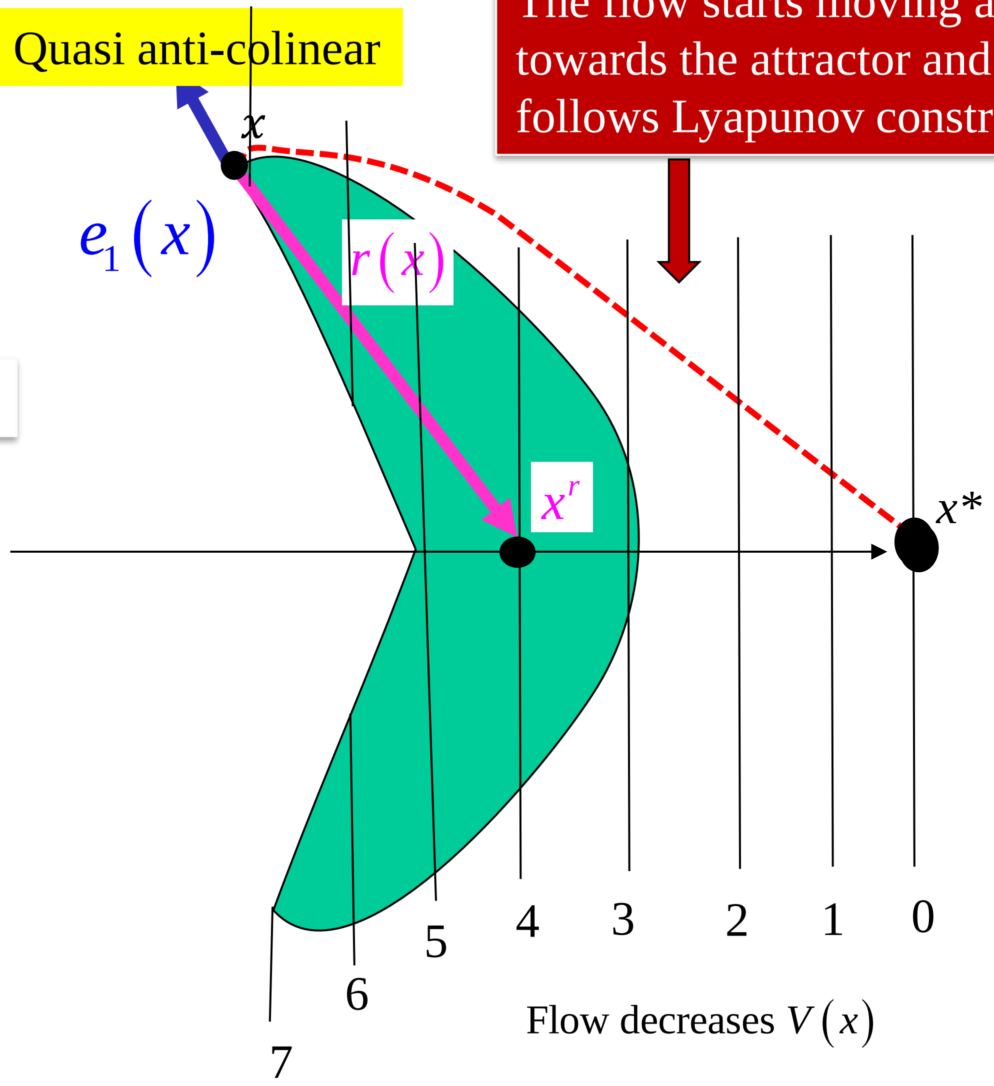
Angle becomes larger, value of the cos decreases

$$P(x) = \theta(x)\theta(x)^T = \begin{bmatrix} r^2(x) & r(x)e_1(x)\cos(\alpha) \\ r(x)e_1(x)\cos(\alpha) & (e_1(x))^2 \end{bmatrix}$$

$$\dot{P}(x) + P(x)\frac{\partial f(x)}{\partial x} + \frac{\partial f(x)}{\partial x}^T P(x) \leq -2\alpha P(x)$$

Temporal Evolution of the Metric

The flow starts moving again towards the attractor and follows Lyapunov constraint.



Metric becomes minimal at the edge of the concavity.

$$P(x)=\theta(x)\theta(x)^T=\begin{bmatrix} r^2(x) & r(x)e_1(x)\cos(\alpha) \\ r(x)e_1(x)\cos(\alpha) & (e_1(x))^2 \end{bmatrix}$$

$$\boxed{\dot{P}(x)+P(x)\frac{\partial f(x)}{\partial x}+\frac{\partial f(x)}{\partial x}^T P(x)\le -2\alpha P(x)}$$

Convergence under Contraction Theory

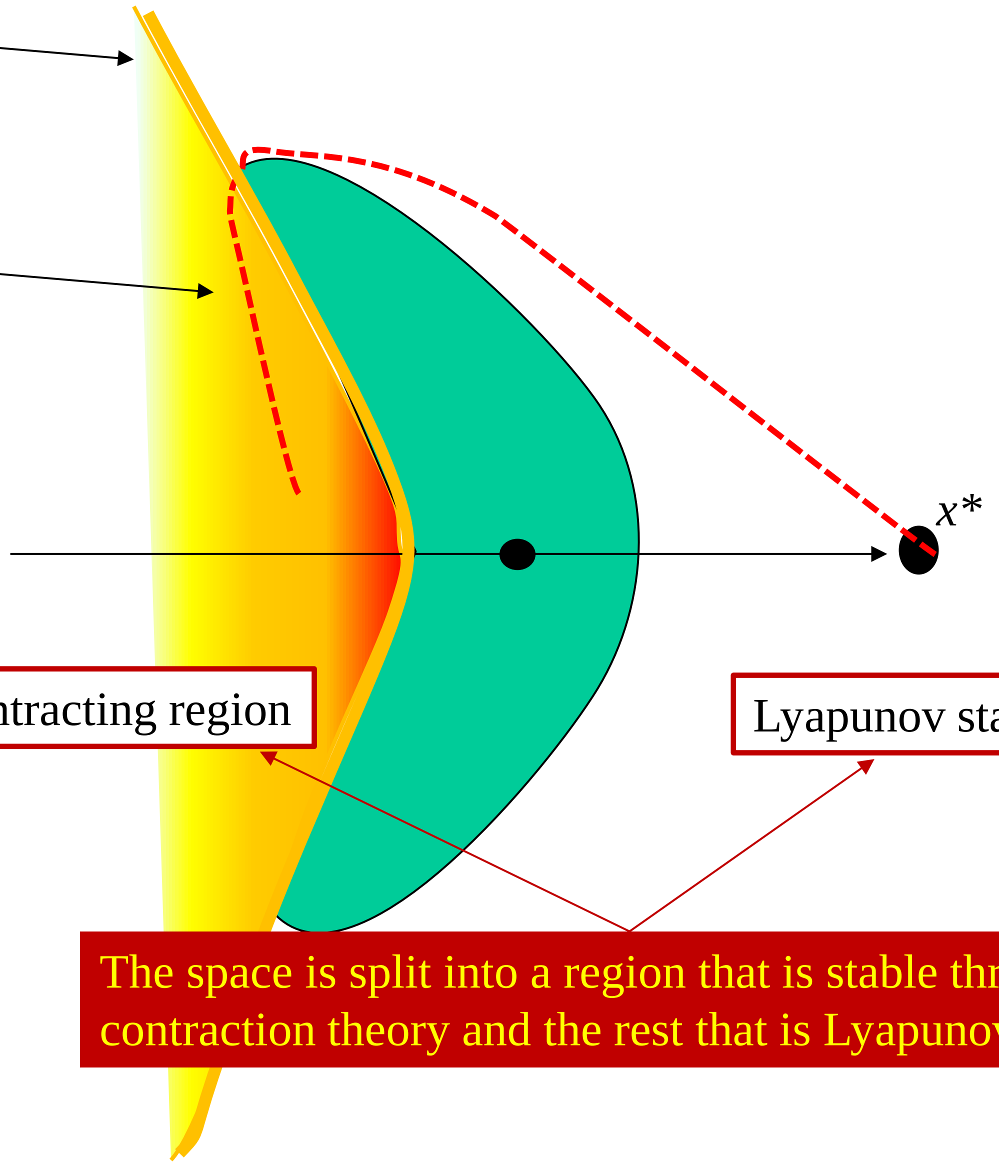
Isoline of the system under the new metric

Modulated flow moves across isolines in such a way that it decreases the metric

Contracting flow

$$\frac{d}{dt} f(x)^T P(x) f(x) < 0$$

$$\dot{P}(x) + P(x)\frac{\partial f(x)}{\partial x} + \frac{\partial f(x)}{\partial x}^T P(x) \leq -2\alpha P(x)$$



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

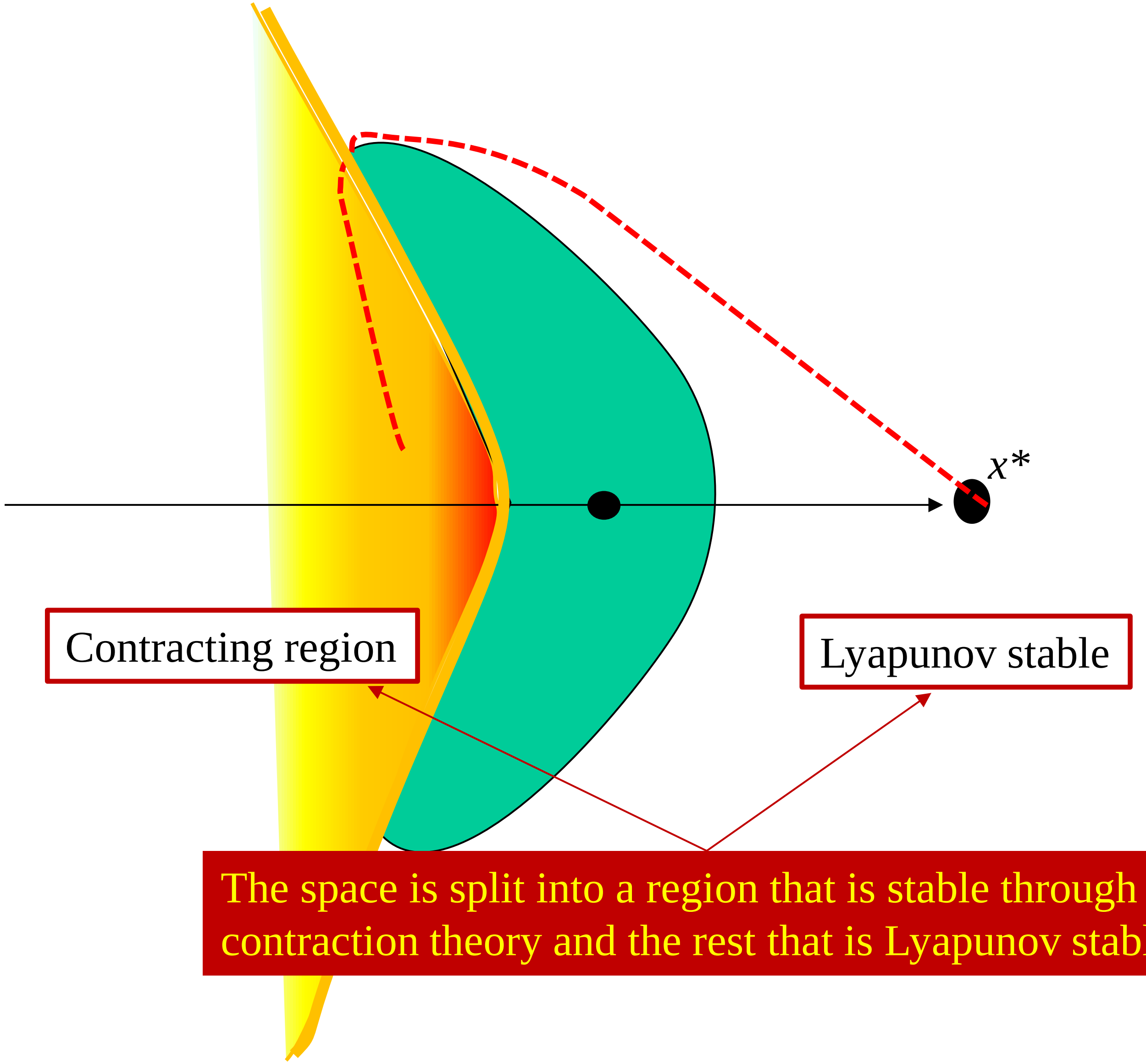
Convergence under Contraction Theory

To prove convergence, we need to prove 2 more things:

1: the system stays away from obstacle for finite time

2: the system never reaches the obstacle's boundary, isoline at which the determinant of the metric is zero

(see full proof in Huber et al 2019 + annexes of book)



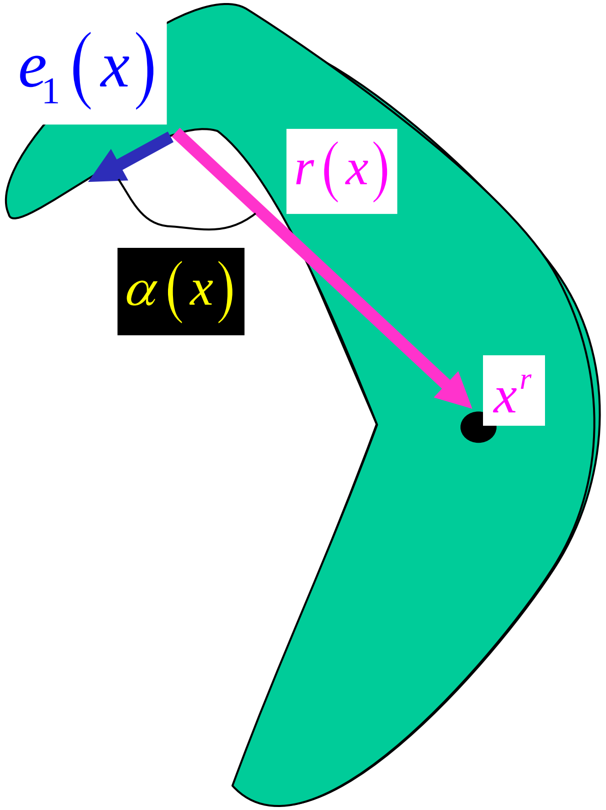
The space is split into a region that is stable through contraction theory and the rest that is Lyapunov stable.

Which concave obstacles?

Can one avoid any concave obstacle with such an approach?

No: the system no longer contracts if the curvature is too large.

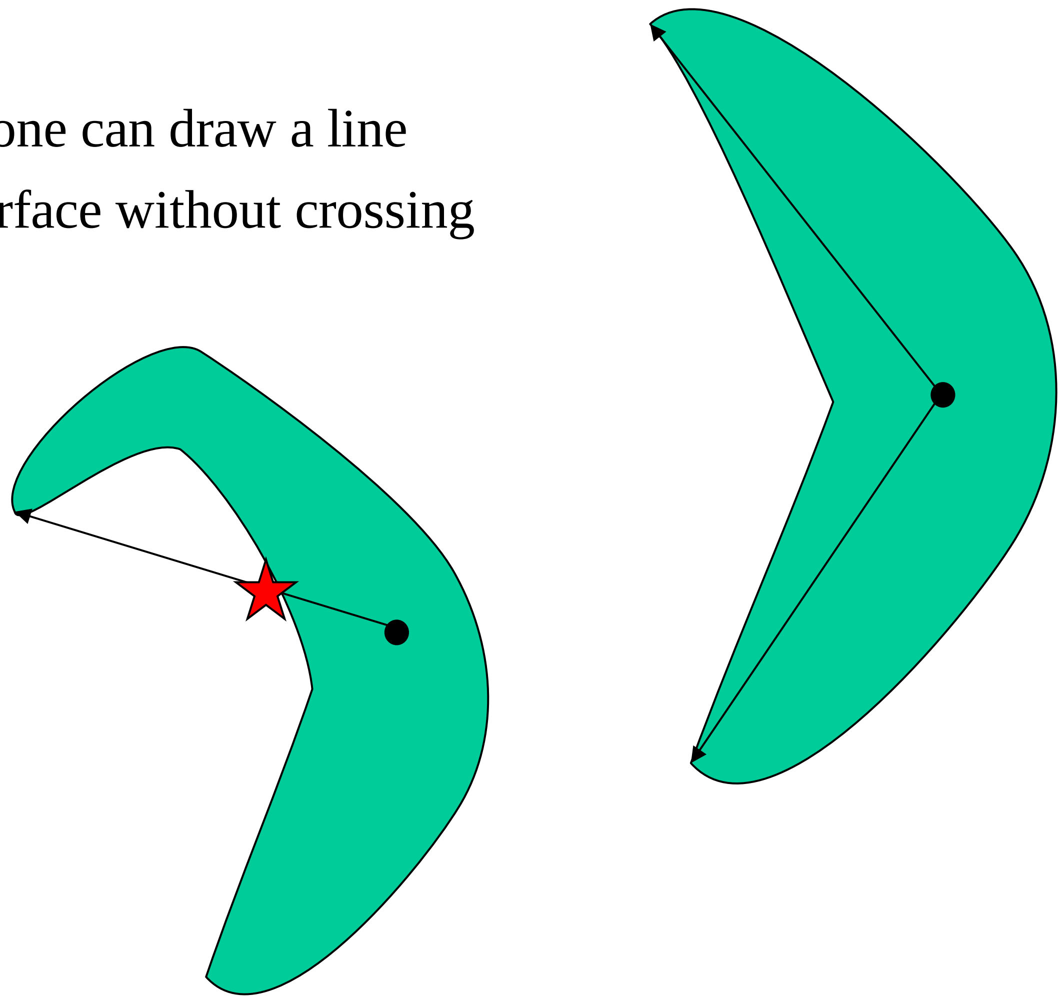
$\alpha(x)$ starts growing



Star-Shaped Concave Obstacles

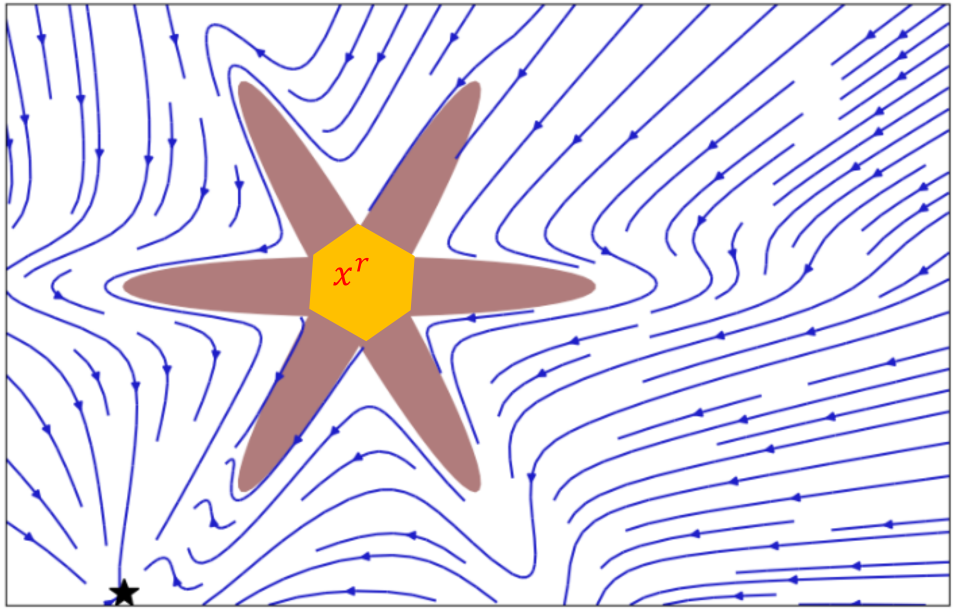
One can avoid any star-shaped concave obstacle.

$\exists$ a point inside the obstacle, from which one can draw a line that reaches any point on the obstacle's surface without crossing the surface.



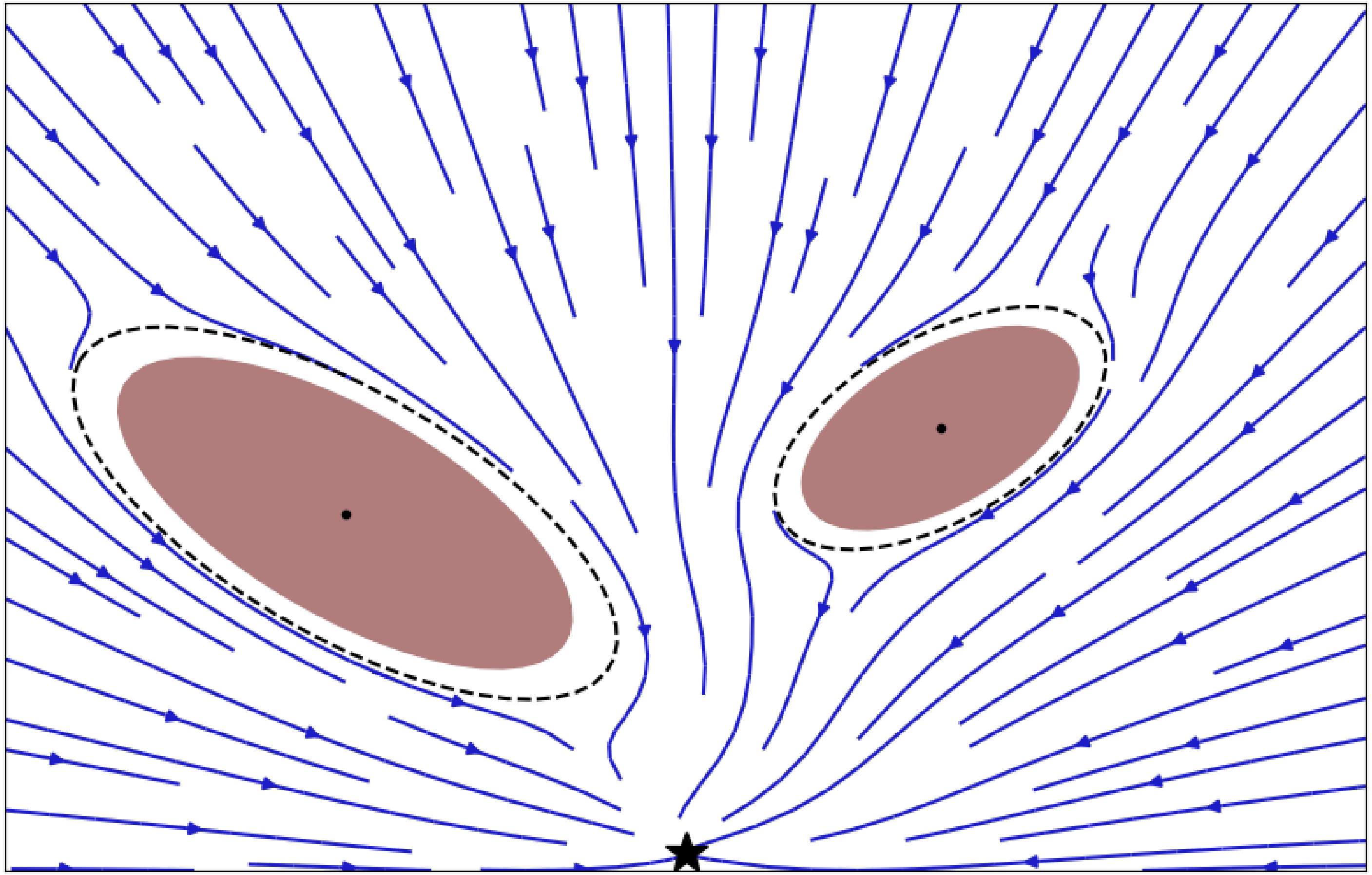
Reference Point in Star-Shaped Concave Obstacles

For star-shaped concave obstacles, the reference point must be located in the region with overlap

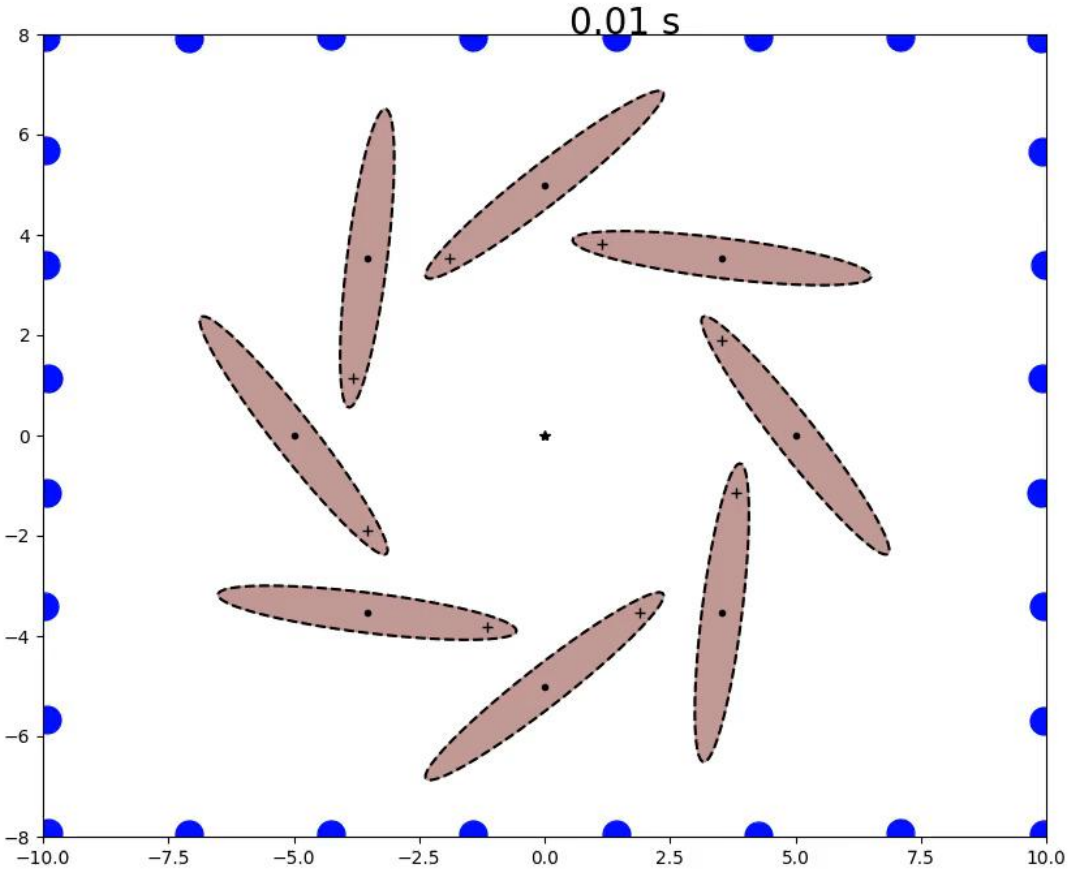


Avoiding Multiple Obstacles

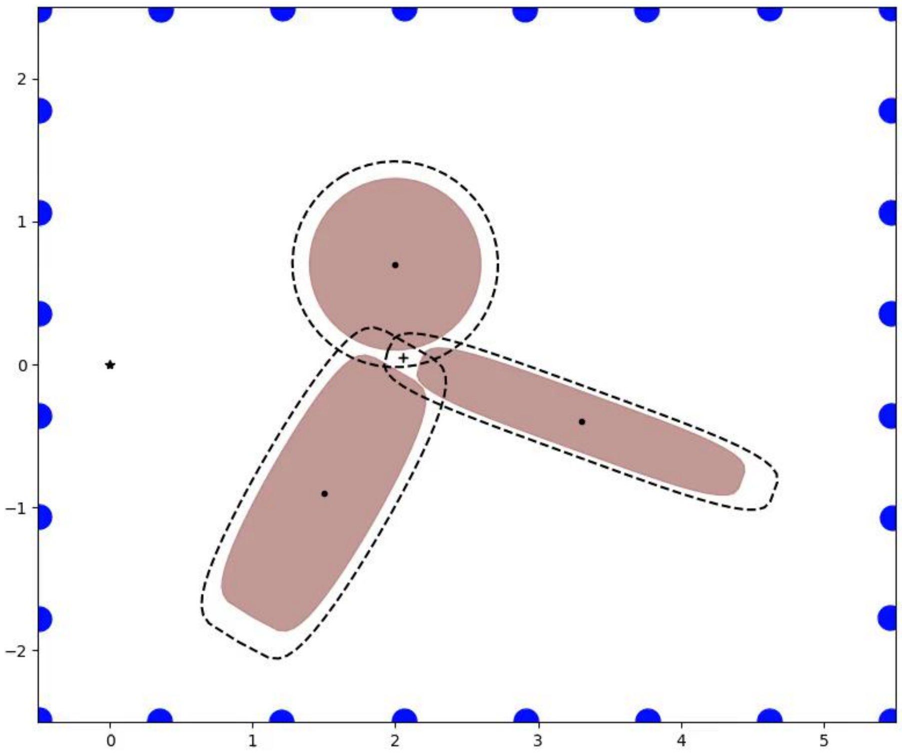
Weighted modulation across the two obstacles – weight the orientation and magnitude of deflection



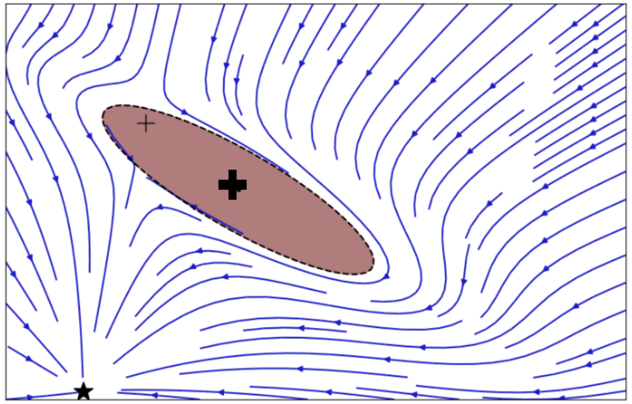
Multiple Obstacles with No Common Points



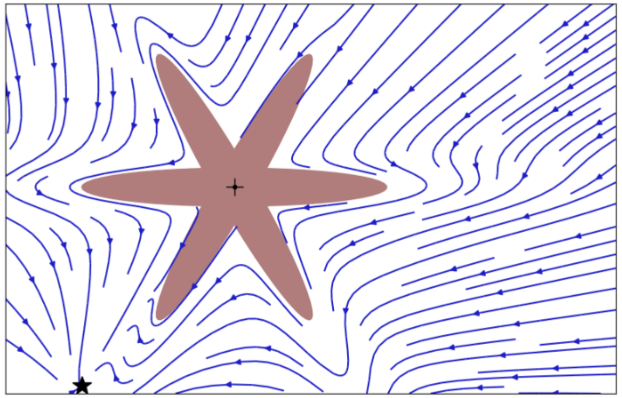
Multiple Obstacles with Common Point



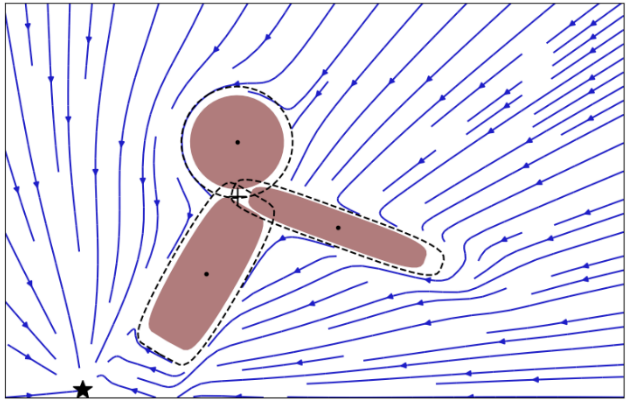
Summary of Avoidance Properties per Type of Obstacles



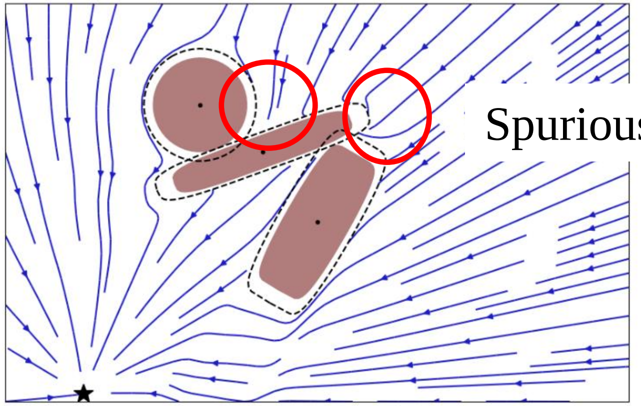
(a) Convex obstacles



(b) Star-shaped obstacles

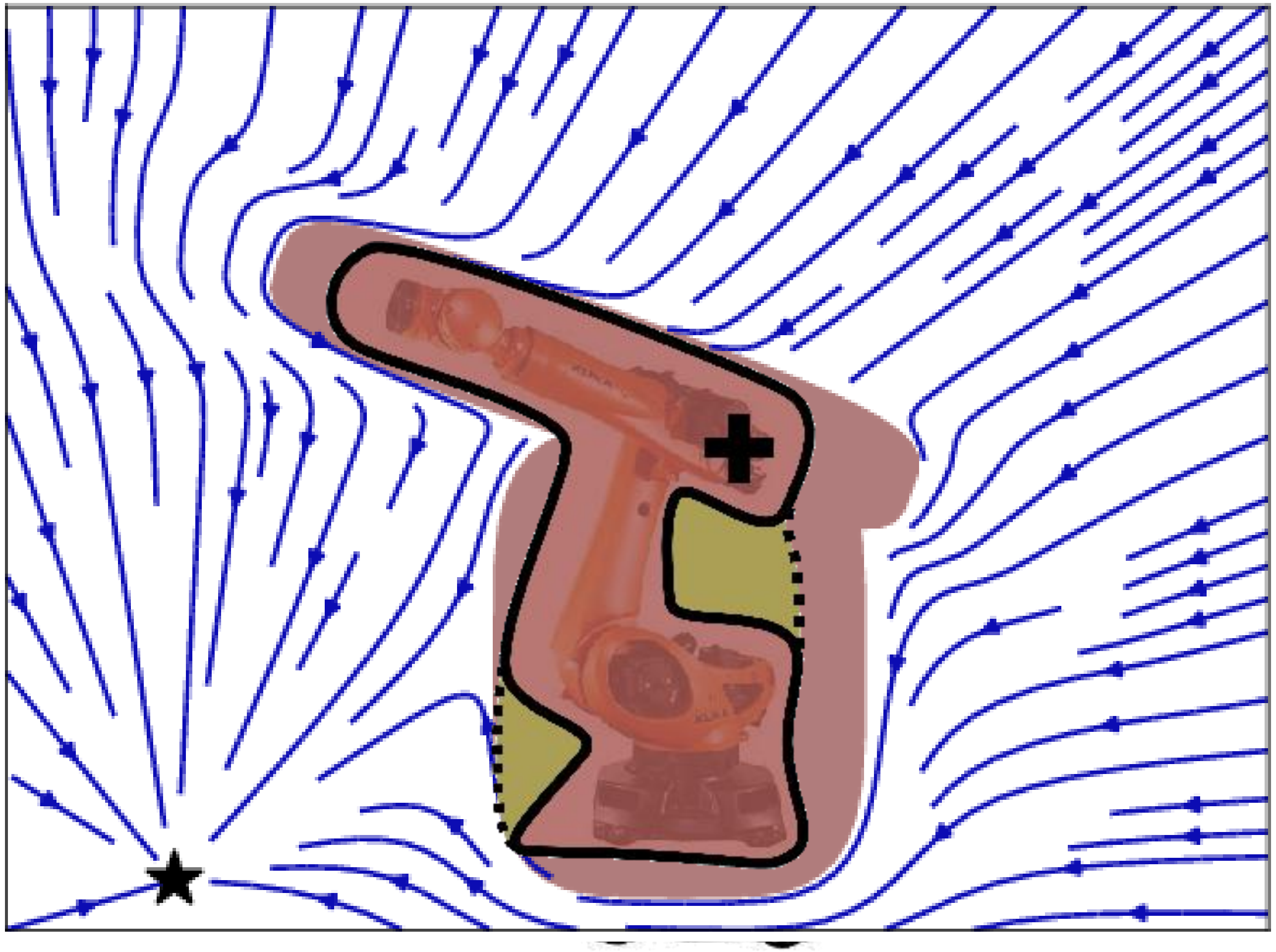


(c) Intersecting convex obstacles with common region



(d) Intersecting obstacles without common region

Making a Non Star-Shaped Obstacle Star-Shaped

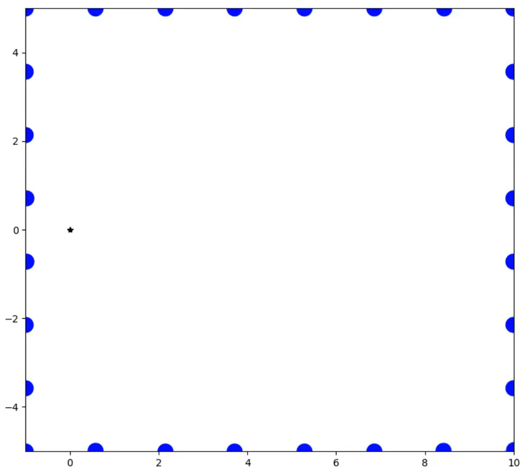


Avoiding Moving Obstacles

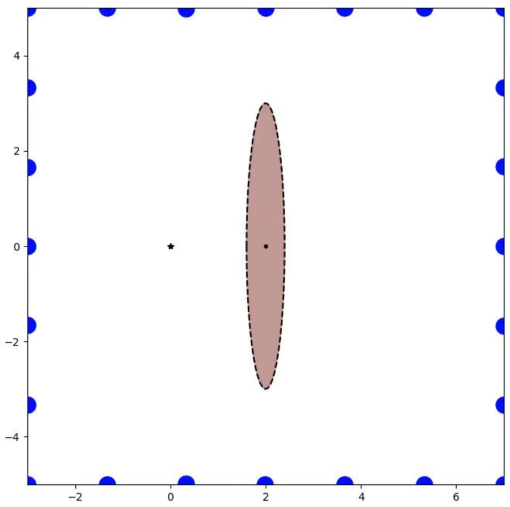
Adapt Local Modulation to Speed of the Obstacle

$$\dot{x} = M(x)\left(f(x) - \dot{x}^o\right) + \dot{x}^o$$

Speed of obstacle

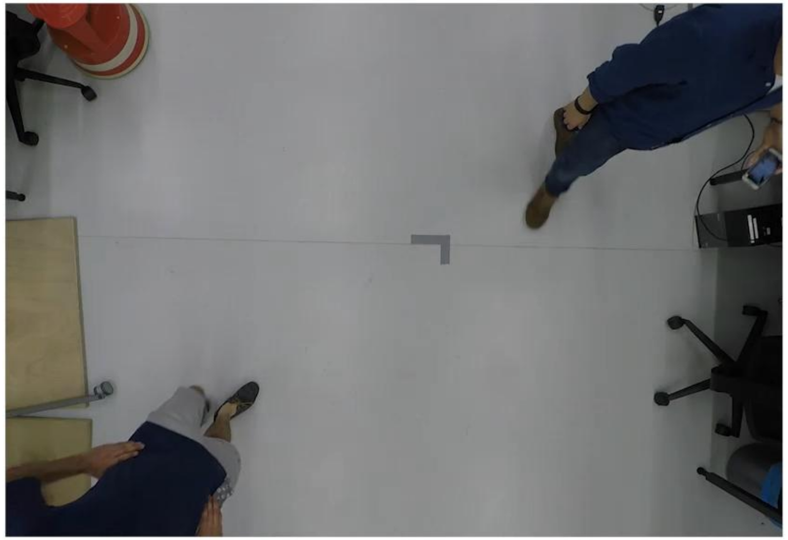


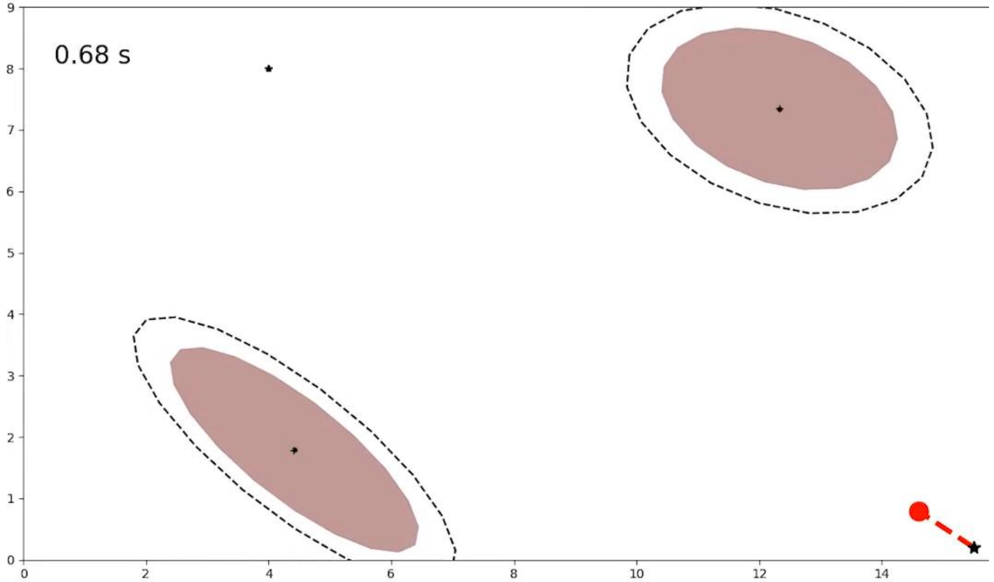
Translation



Rotation

Avoiding Moving Obstacles

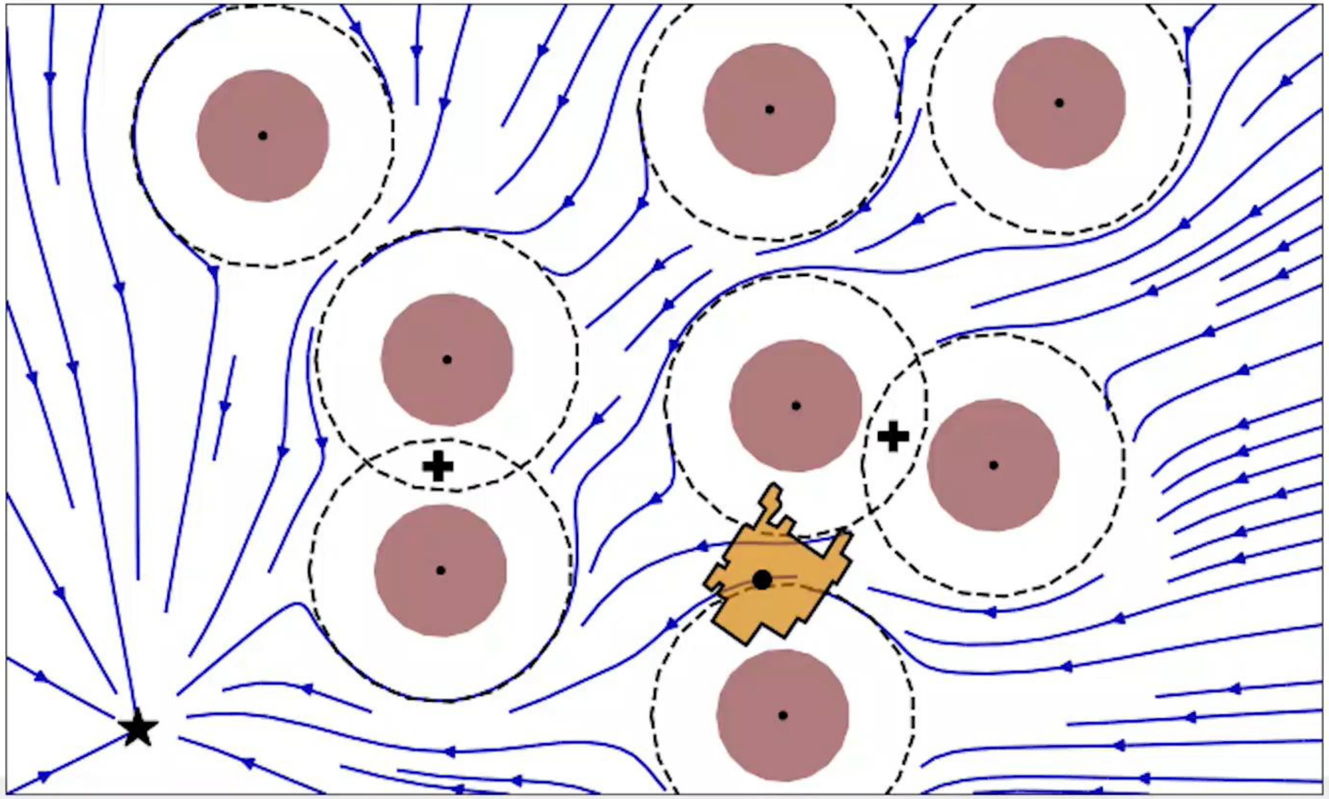




Two persons can be interpreted as ellipses, while a hypothetical-point robot is trying to avoid them.

Modulation is applied partially in moving frame to avoid penetration of boundary, but also keeping attractor position close to the original one.

Live Obstacle Avoidance with a Simulated Wheelchair



Humans are represented by spheres and have a safety margin to account for the wheelchair's geometry.

This work was conducted partly under a grant from the EU commission through the Crowdbot project.

Live Obstacle Avoidance in a simulated and Real Crowd

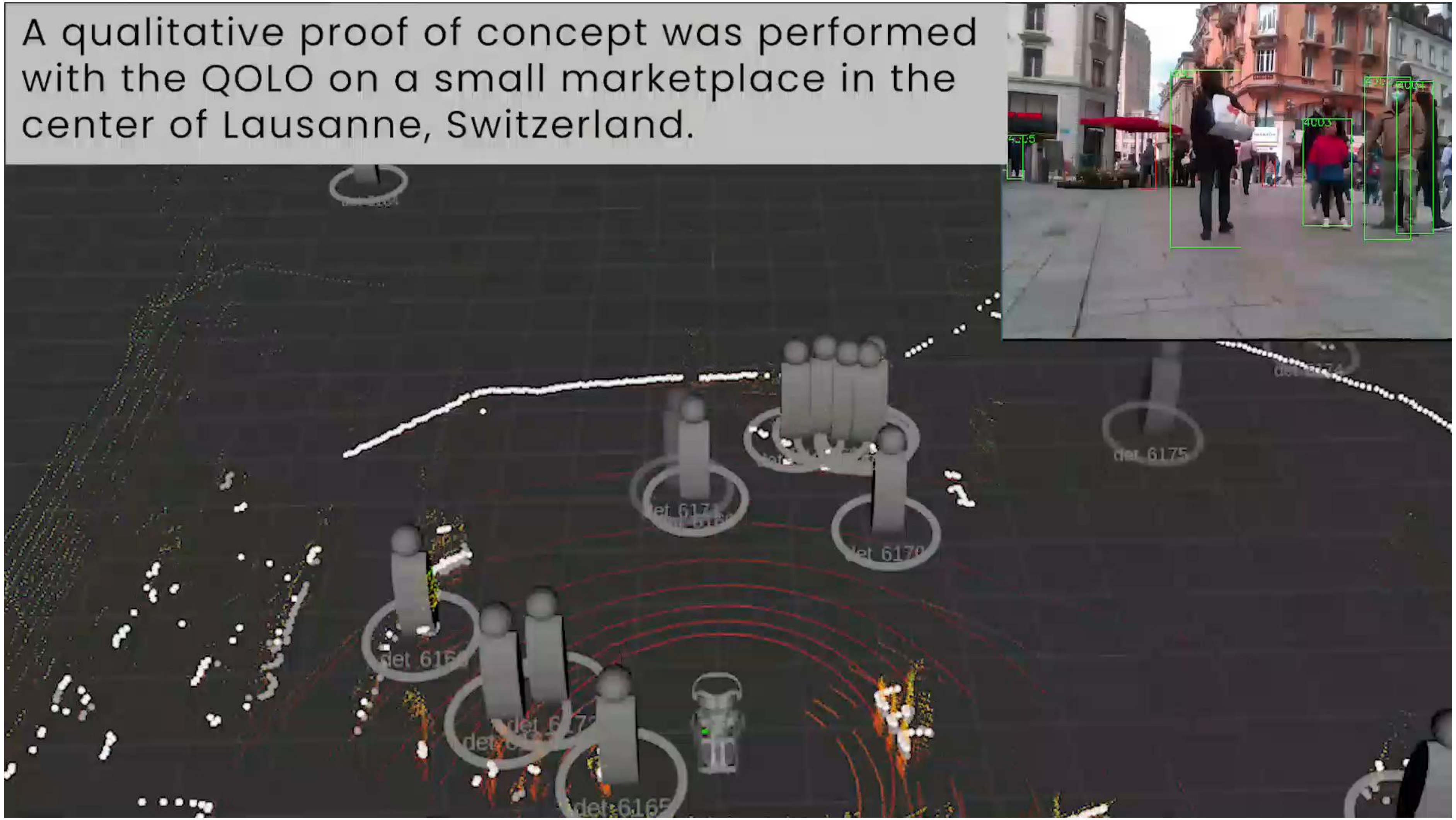
The experimental evaluation is done on the standing wheelchair QOLO [4]. The wheelchair behaves as an autnomous agent in the crowd-simulator developped by [5].

QOLO [4]

Crowd-simulator [5]

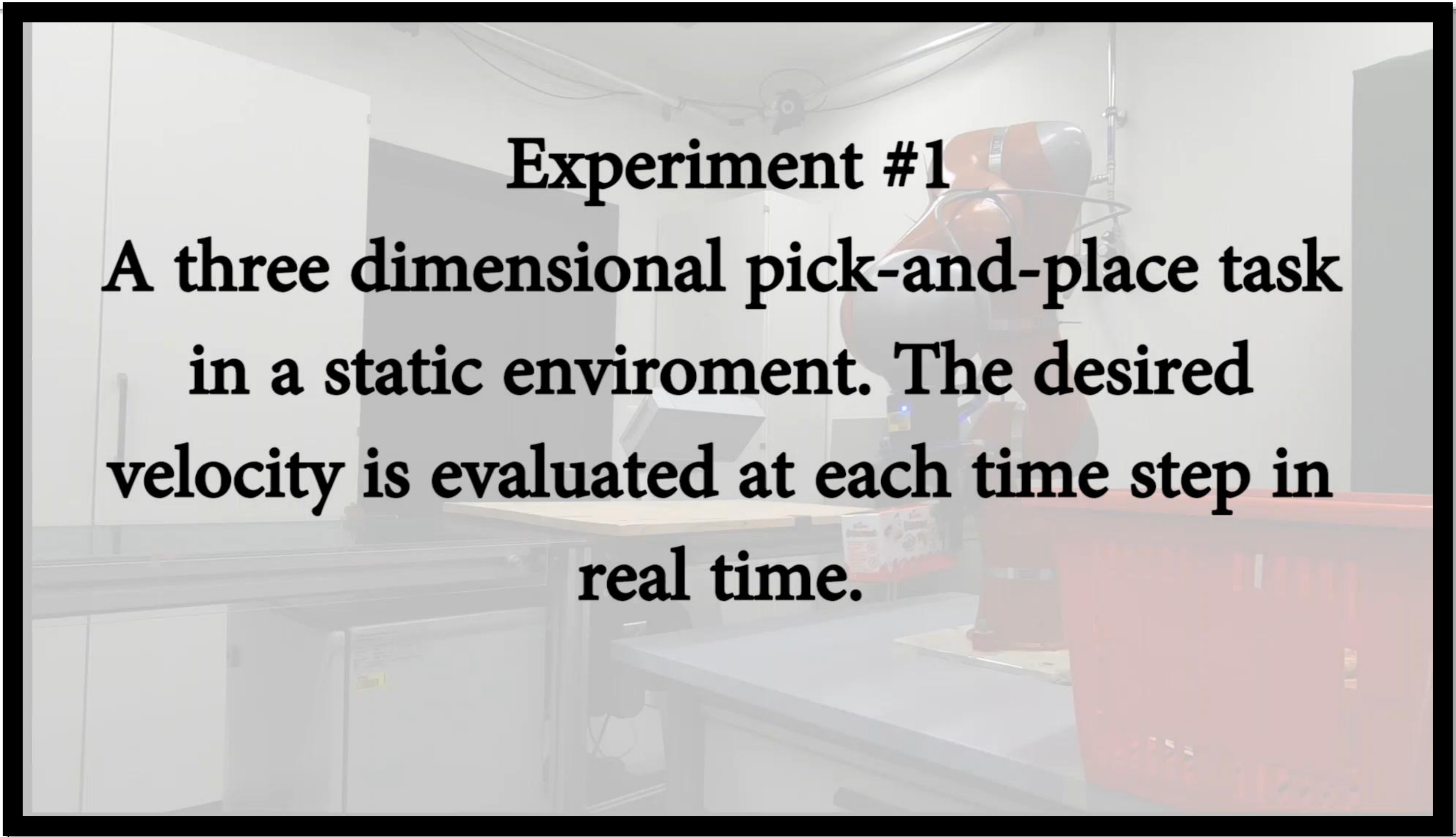
[4] Granados, Diego Felipe Paez, Hideki Kadone, and Kenji Suzuki. "Unpowered Lower-Body Exoskeleton with Torso Lifting Mechanism for Supporting Sit-to-Stand Transitions." 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018.
[5] F. Grzeskowiak, M. Babel, J. Bruneau, and J. Pettre, "Toward virtual reality-based evaluation of robot navigation among people," in IEEE VR 2020-27th IEEE Conf. on Virtual Reality and 3D User Interfaces, 2020, pp. 1–9.

Live Obstacle Avoidance in a simulated and Real Crowd



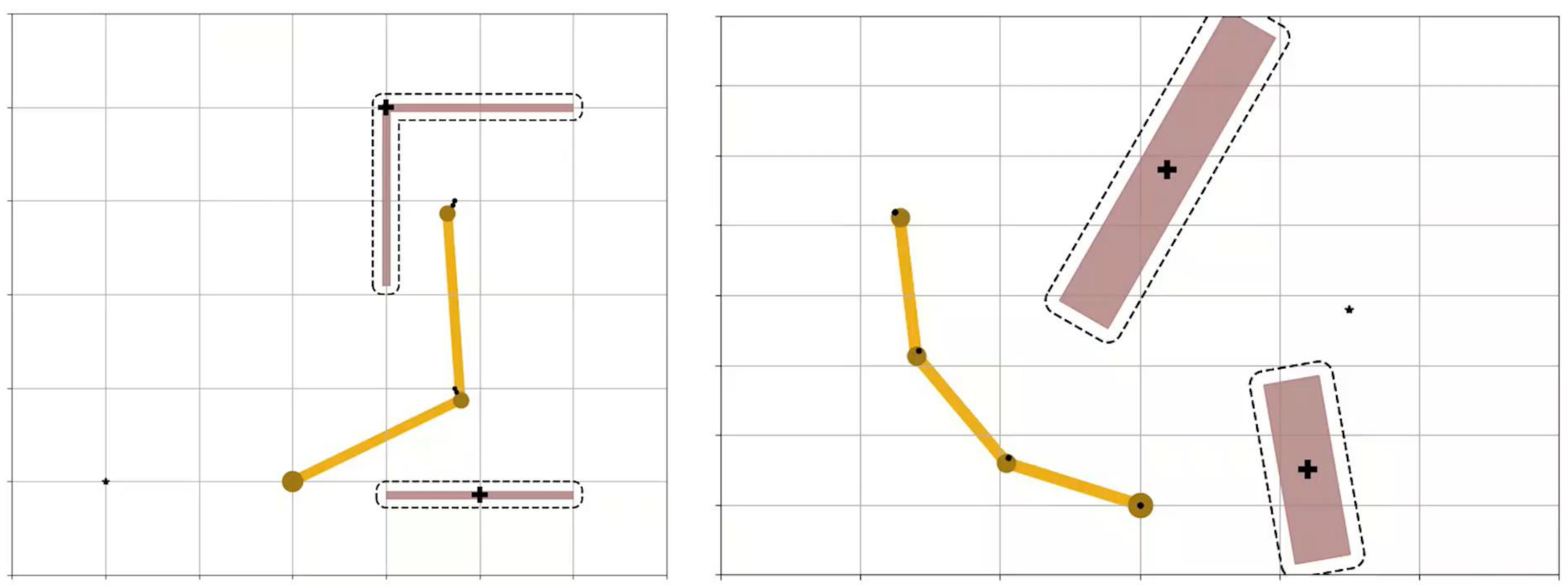
Huber, Lukas, Jean-Jacques Slotine, and Aude Billard. "Avoiding Dense and Dynamic Obstacles in Enclosed Spaces: Application to Moving in Crowds. IEEE Transactions on Robotics, 2022.

Live Obstacle Avoidance in Task Space



Huber, Lukas, Aude Billard, and Jean-Jacques Slotine. "Avoidance of convex and concave obstacles with convergence ensured through contraction." *IEEE Robotics and Automation Letters* 4.2 (2019): 1462-1469.

Live Obstacle Avoidance in Joint Space



The method is evaluated along the robot arm to ensure collision avoidance. The effect is weighted based on the danger-field (distance field).

Huber, Lukas, Jean-Jacques Slotine, and Aude Billard. "Avoiding Dense and Dynamic Obstacles in Enclosed Spaces: Application to Moving in Crowds. IEEE Transactions on Robotics, 2022.

Enclosing the dynamics

The obstacle's boundary can also be used as an enclosing; once inside, the flow cannot escape. This can be achieved with a simple inversion of the distance function. The new distance function is $1/\Gamma$. The normal is obtained by flipping the current robot state x along the reference direction

Modulation is given by:

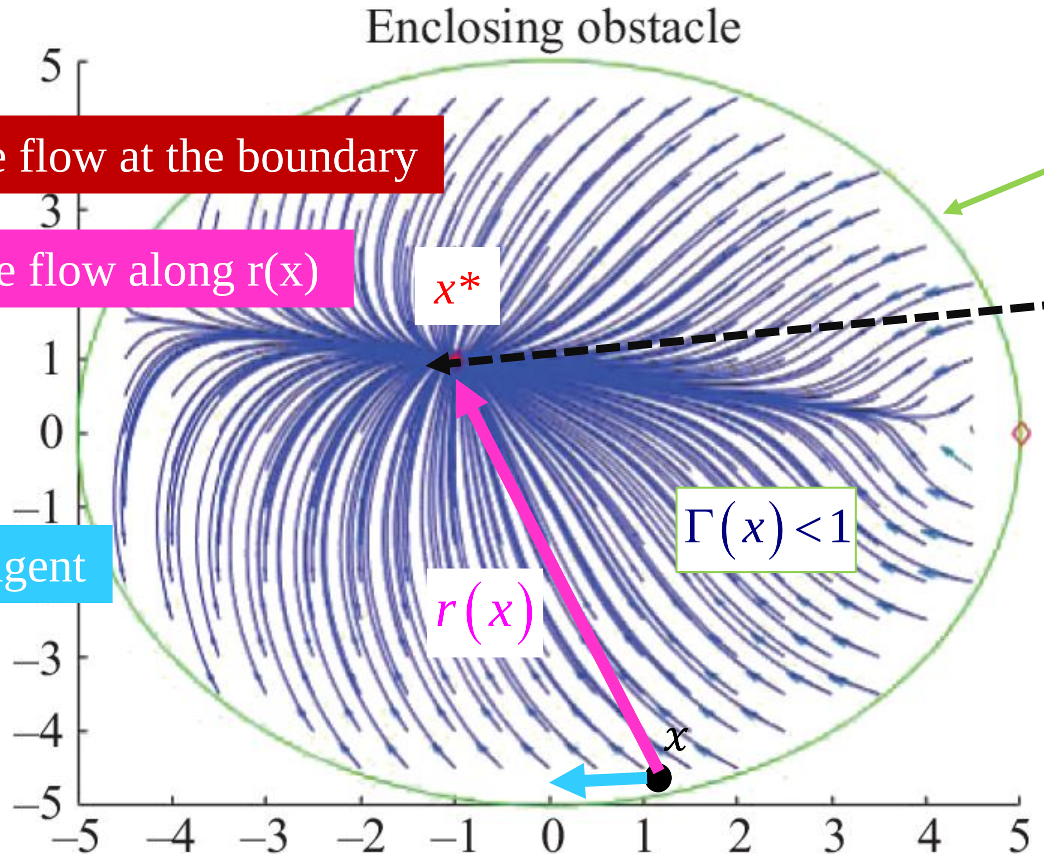
$$\lambda_1(\tilde{x}) = 1 - \frac{1}{\Gamma(\tilde{x})}$$

Cancels the flow at the boundary

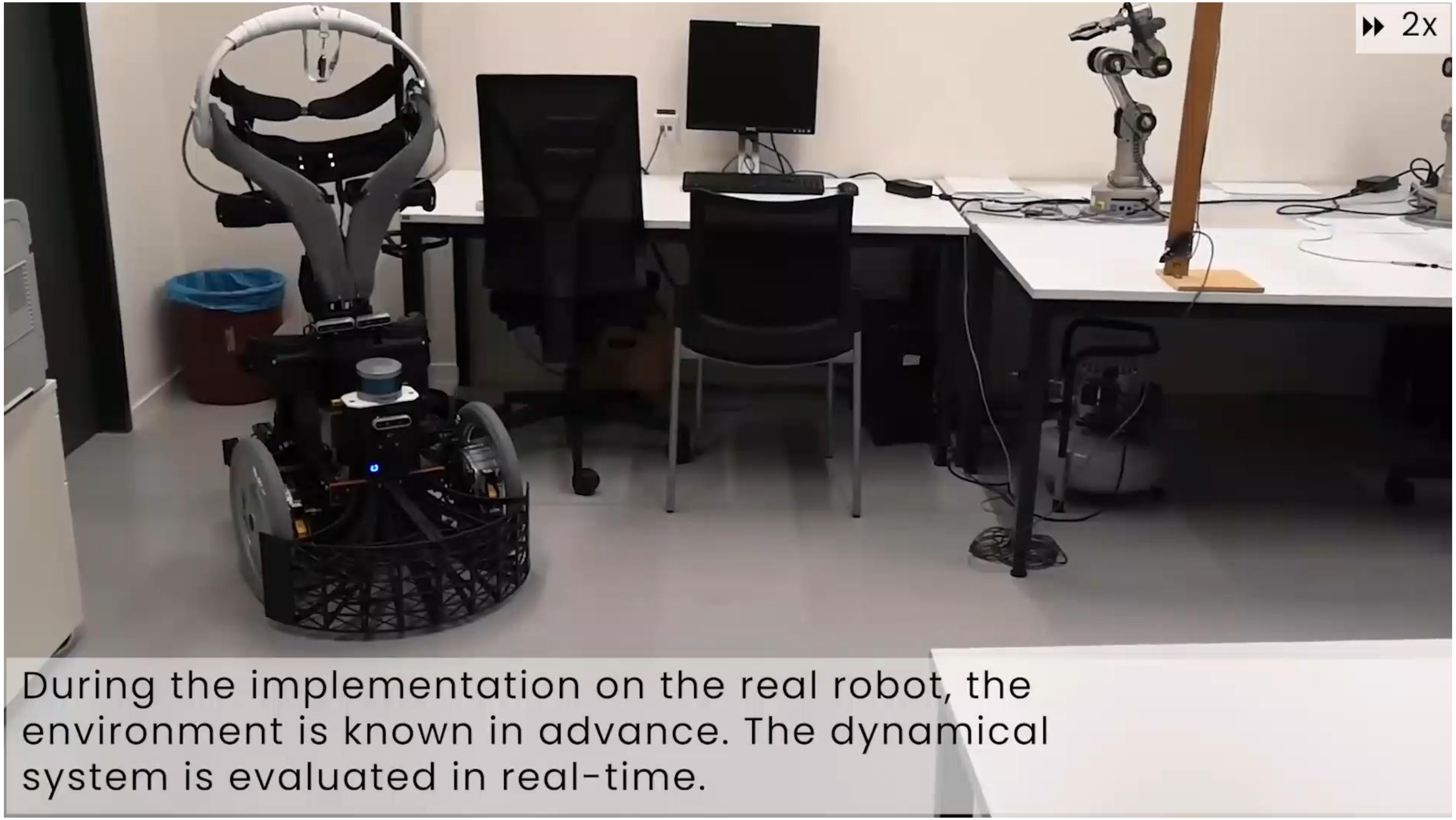
Reduces the flow along r(x)

$$\lambda_2(\tilde{x}) = 1 + \frac{1}{\Gamma(\tilde{x})}$$

Redirects the flow along the tangent



Live Obstacle Avoidance in a simulated and Real Crowd



Huber, Lukas, Jean-Jacques Slotine, and Aude Billard. "Avoiding Dense and Dynamic Obstacles in Enclosed Spaces: Application to Moving in Crowds. IEEE Transactions on Robotics, 2022.

SUMMARY – 2nd Part

- The obstacle modulation approach has been extended to enable obstacle avoidance of star-shaped concave obstacles by *modifying the basis for a non-orthonormal basis.*
- This change of coordinate allowed us to define a new metric in space, *under which the system is contracting.*
  → Convergence to the attractor can again be ensured globally through *contraction theory*.
- The approach can be extended to handle *multiple and moving obstacles*.
- The approach can also allow *inverting obstacle avoidance and ensure the robot does not escape a region*.

Limitations

- The approach requires the *obstacle to be star-shaped*. Any obstacle can be converted into a star-shaped one at the cost of loosing free space.
- The inverted obstacle is also limited to star-shaped volumes.
- The approach retains *one spurious fixed point*, but this is a saddle point and only a unique trajectory leads to this point. In practice, this can be avoided easily.

Self-Collision, Joint-Level Obstacle Avoidance

- Learn the boundaries of arbitrary complex concave obstacles.
    - A robot or group of robots' joint workspace is an example of such complex boundary.
- Develop a method to enable live-obstacle avoidance in this complex boundary.

- Figueroa Fernandez, Nadia Barbara, Seyed Sina Mirrazavi Salehian, and Aude Billard. "Multi-Arm Self-Collision Avoidance: A Sparse Solution for a Big Data Problem." *Proceedings of the Third Machine Learning in Planning and Control of Robot Motion (MLPC) Workshop.*
- Mirrazavi Salehian, Seyed Sina, Nadia Figueroa, and Aude Billard. "A unified framework for coordinated multi-arm motion planning." *The International Journal of Robotics Research* 37.10 (2018): 1205-1232.

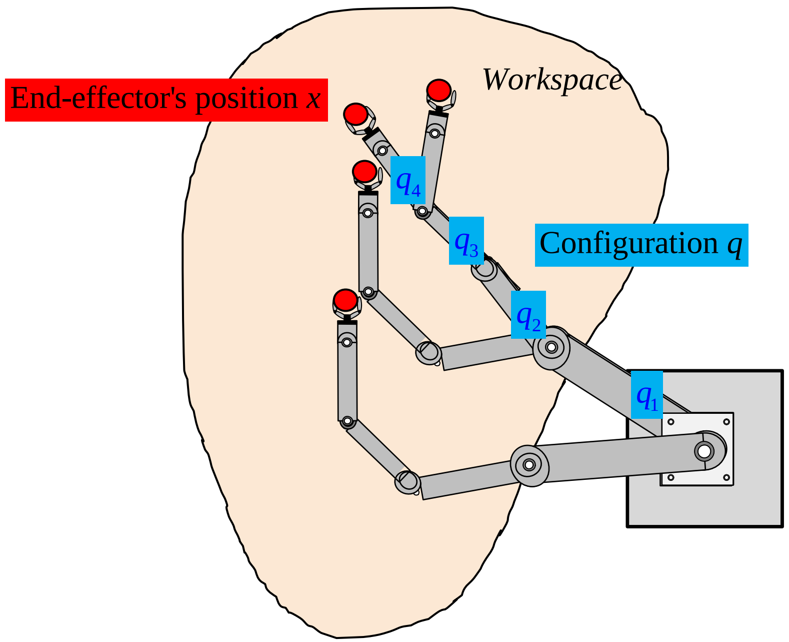
Configuration space versus task space

- The feasible space of motion of the *joints* is called the *configuration space*.
- The feasible space of motion of the robot's end-effector in Cartesian space is called the *workspace*.

Configuration space: C: $\left\{ q \in \mathbb{R}^{N_q} ; q_i^{\min} \leq q_i \leq q_i^{\max} \right\}$

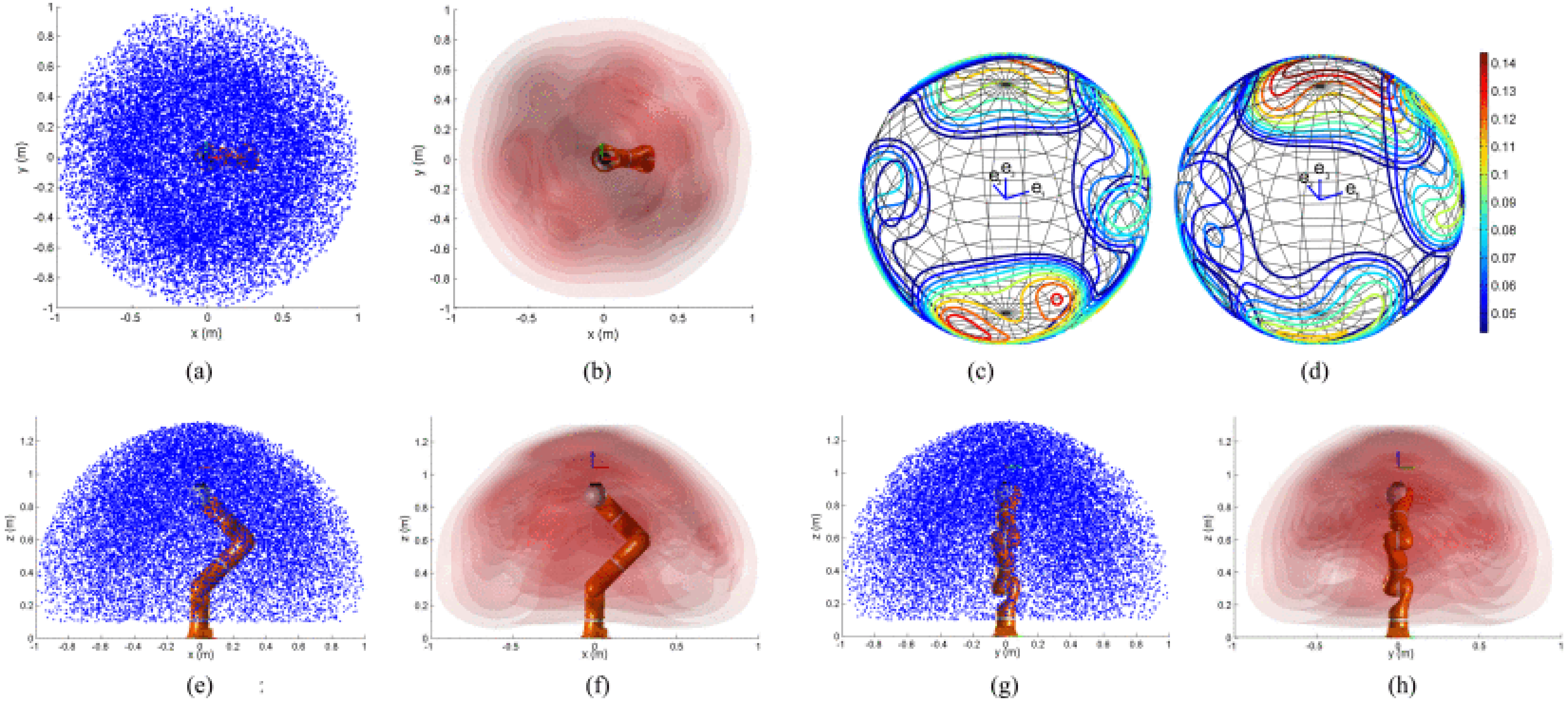
Workspace: W: $\left\{ x \in \mathbb{R}^{N_x} ; \exists q, \text{ s.t. } x = h(q) \right\}$: h: forward kinematics

Usually, N_q =7 and N_x =3 or N_x = 6.



Learning a Model of the Configuration space

The configuration space, or C-space, of the robot system is the space of all possible configurations of the system.

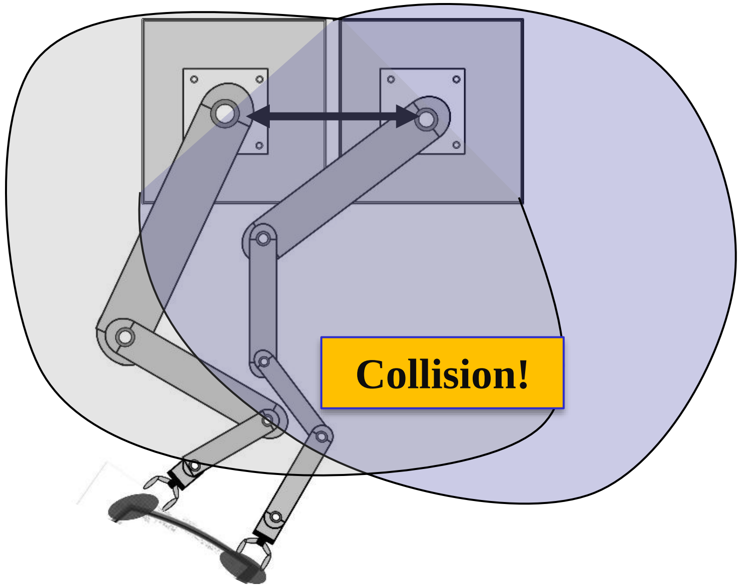


For a robot whose kinematic chain is known, one can sample the space and learn a model of the configuration space, as a distribution of joint configuration. Above: a model for the 7 DOFs KUKA LWR arm has been learned using Gaussian Mixture Model.

Kim , Shukla and Billard, " IEEE Transactions on Robotics 30, no. 5 (2014): 1049-1065.

Learning a model of free vs collided configuration space

In a bimanual system, the free space is not fixed.
Free space for each of the arm varies as a function of the motion of the other arm



Define a measure of distance to the boundary across free and collided space

$$\Gamma(q^{ij}) : \mathbb{R}^{d_{q_i}+d_{q_j}} \rightarrow \mathbb{R}$$

$$q^{ij} = [q^i, q^j]^T \in \mathbb{R}^{d_{q_i}+d_{q_j}}$$

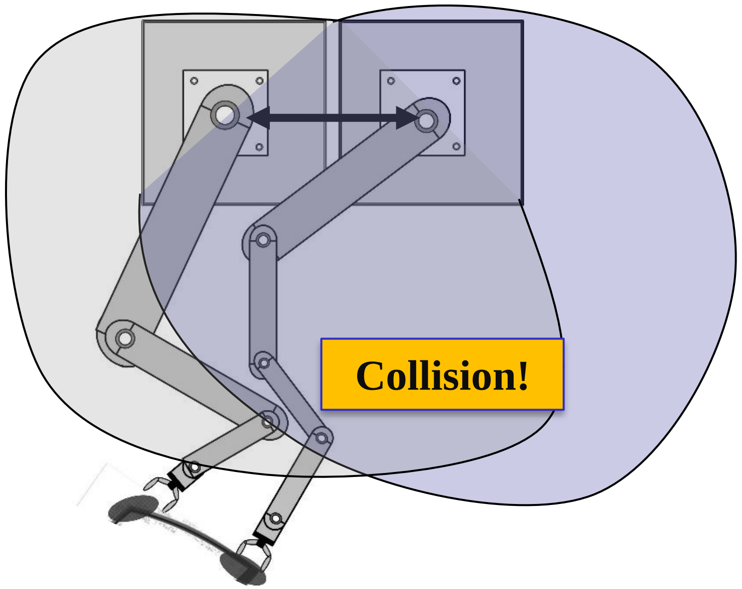
Collided configurations: $\Gamma(q^{ij}) < 1$

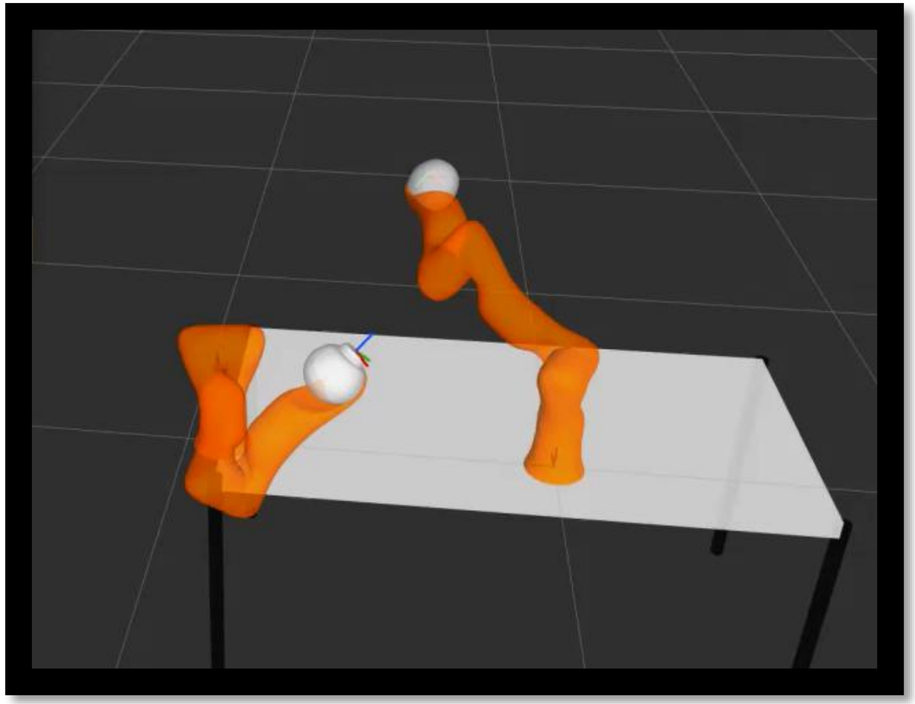
Boundary configurations: $\Gamma(q^{ij}) = 1$

Free configurations: $\Gamma(q^{ij}) > 1$

Mirrazavi, Figueroa and Billard, Int. Journal of Robotics Research, IJRR, 2018

Learning a model of free vs collided configuration space





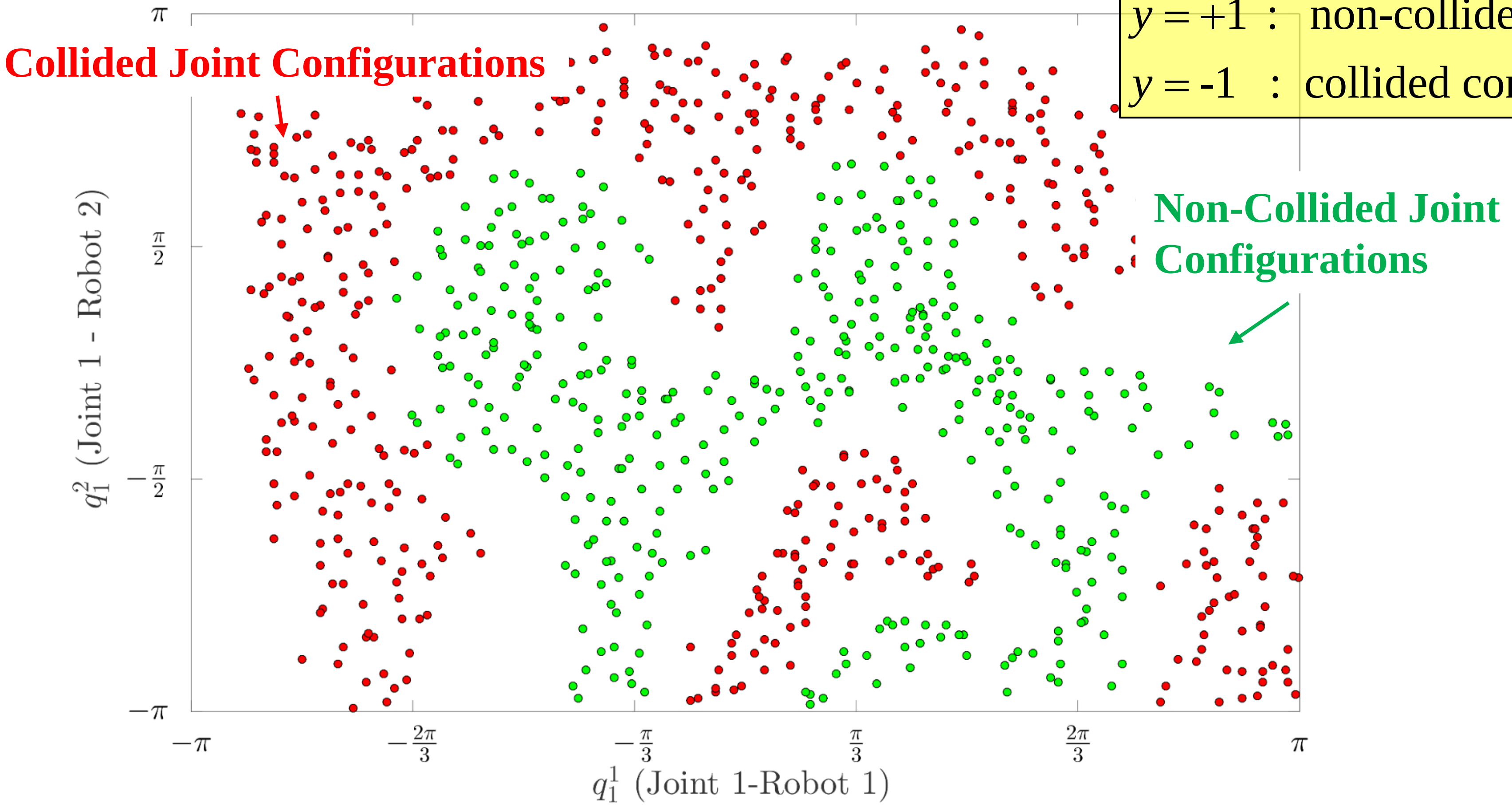
For a robot whose kinematics is known, it is possible to sample the space
And to build a model of the configuration space.

Learning the boundary between free space and collision space
is modelled as a **binary classification problem**

Using Support Vector Classification:

$y = +1$: non-collided configurations

$y = -1$: collided configurations



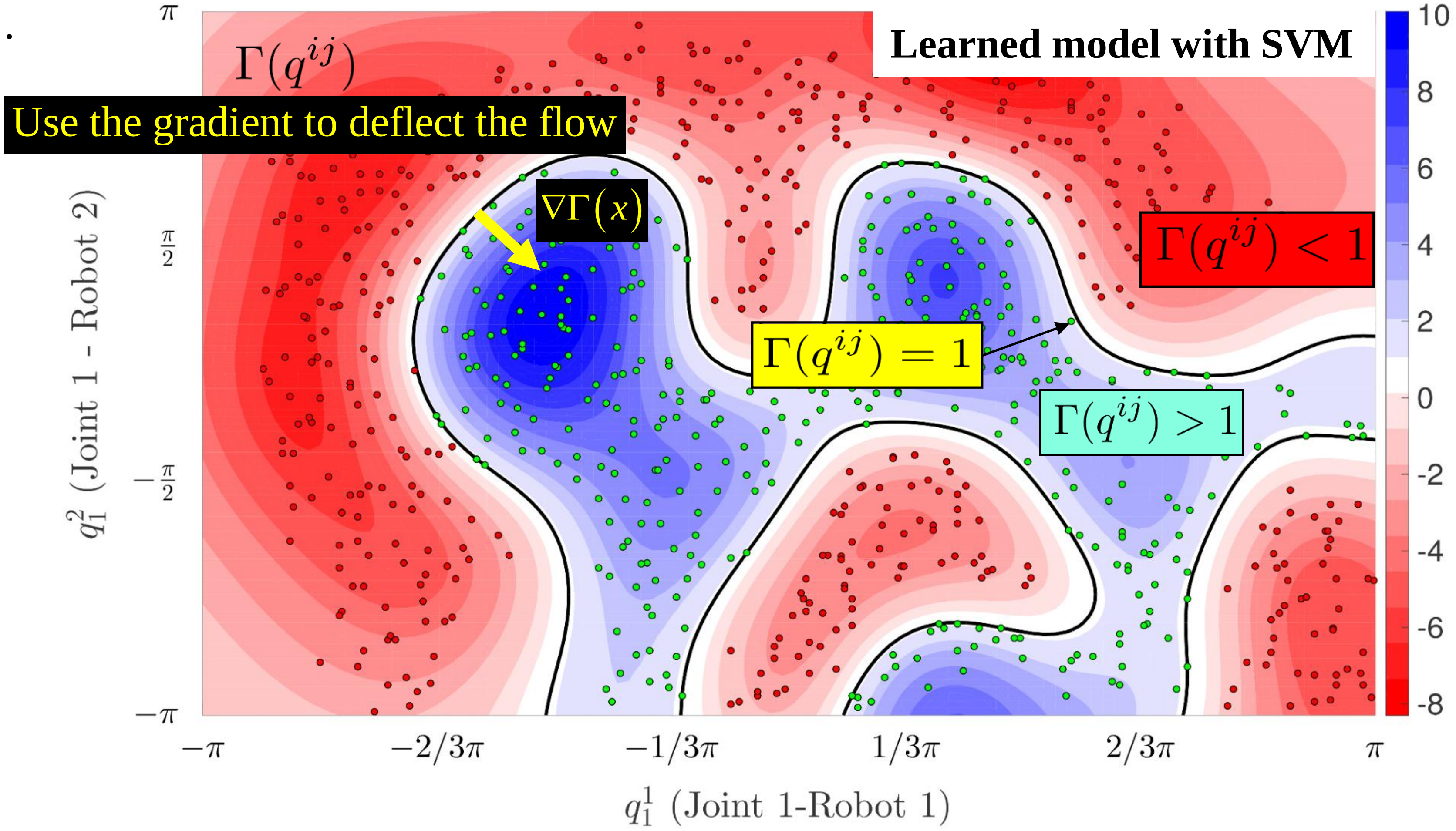
From learned classification to creating the boundary

At testing, select a query joint configuration q^* and apply Support Vector Classification decision function

$y = \mathrm{sgn}\left(h\left(q^*\right)\right).$

Offset to get boundary:

$\Gamma(q) = h(q) + 1.$



Real-Time Control through Local Optimization

Formulate as a Quadratic Programming (QP) problem solvable at run time

Objective function:

$$\underbrace{\arg\min_{\dot{\mathbf{q}}} \frac{\dot{\mathbf{q}}^T W \dot{\mathbf{q}}}{2}}_{\text{Minimize expenditure}} \qquad W \succ 0 \text{ weight matrix}$$

Relax constraint to follow path accurately

Constraints:

$J(q)\dot{q} = \dot{x}$ Inv. Kin. - joint velocities are feasible and lead to the desired path given by some nominal system in Cartesian space $\dot{\mathrm{x}} = f_x(x)$

The two constraints may conflict.

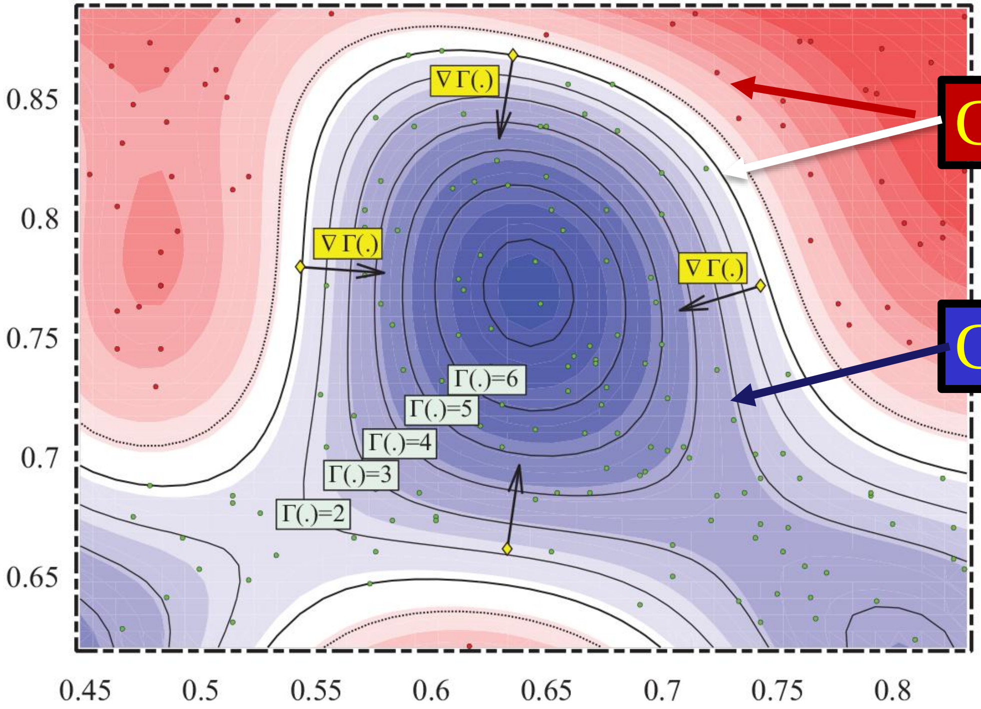
$\dot{\theta}^- \leq \dot{q} \leq \dot{\theta}^-$ Velocities remains within robot's bounds

$$-\nabla\Gamma^{ij}(q^{ij})^T \dot{q}^{ij} \leq \log(\Gamma^{ij}(q^{ij}) - 1)$$

$\forall$ the robots' joints $\dot{q}^{ij}$

Motion will not penetrate the boundary

Constraints on Motion in Configuration Space



Constraint active

Constraint inactive

Convex quadratic program with inequality constraints

Solvable in real-time, but this sets constraints on learning of Γ.

$$-\nabla\Gamma^{ij}(q^{ij})^T\dot{q}^{ij} \leq \log(\Gamma^{ij}(q^{ij})-1)$$

$\forall$ the robots' joints $\dot{q}^{ij}$

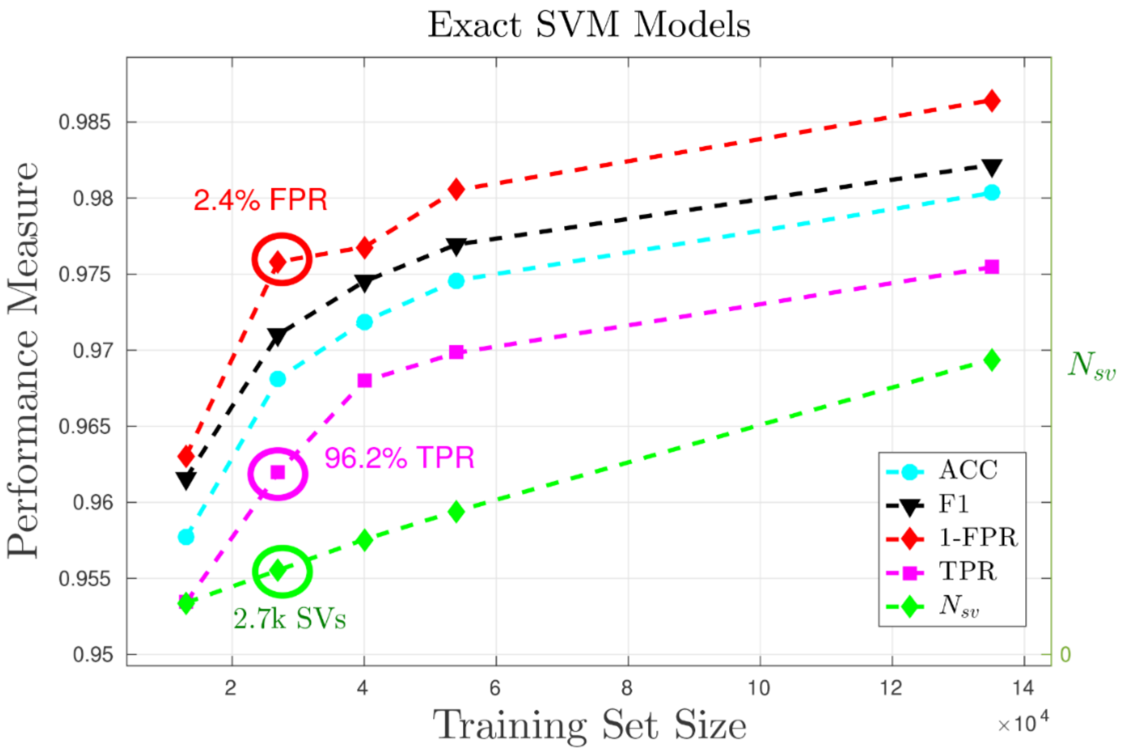
Motion will not penetrate the boundary

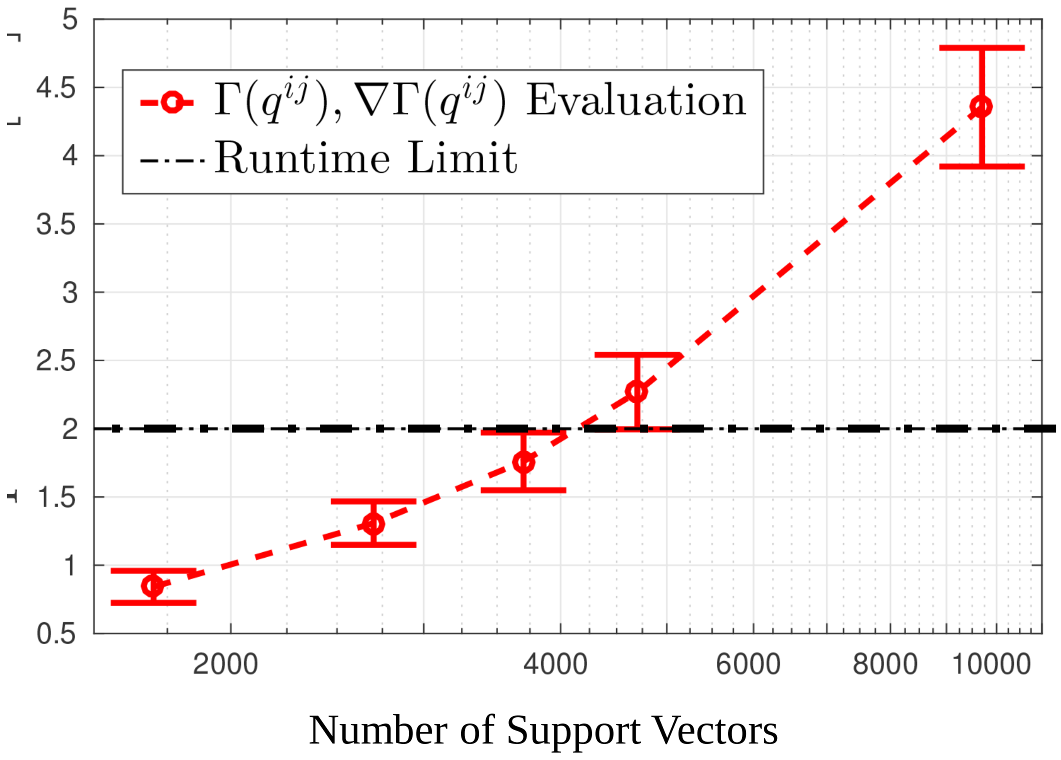
Constraints on Learning Problem

With a base offset of 1.3m we get a dataset of 5.4 million points!
2.4 million points for y=-1 (collision)
3 million points for y=+1 (non-collision)

Force the dataset to be balanced!

The proposed QP Solver needs to run at <2ms!

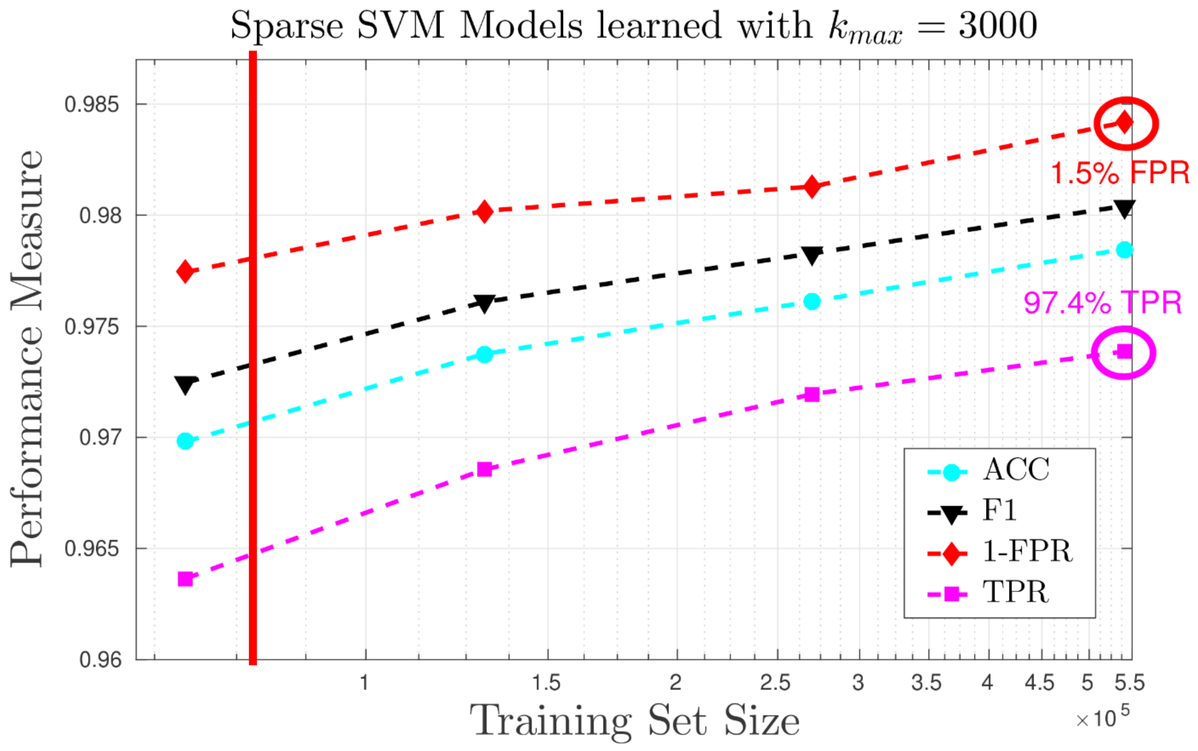


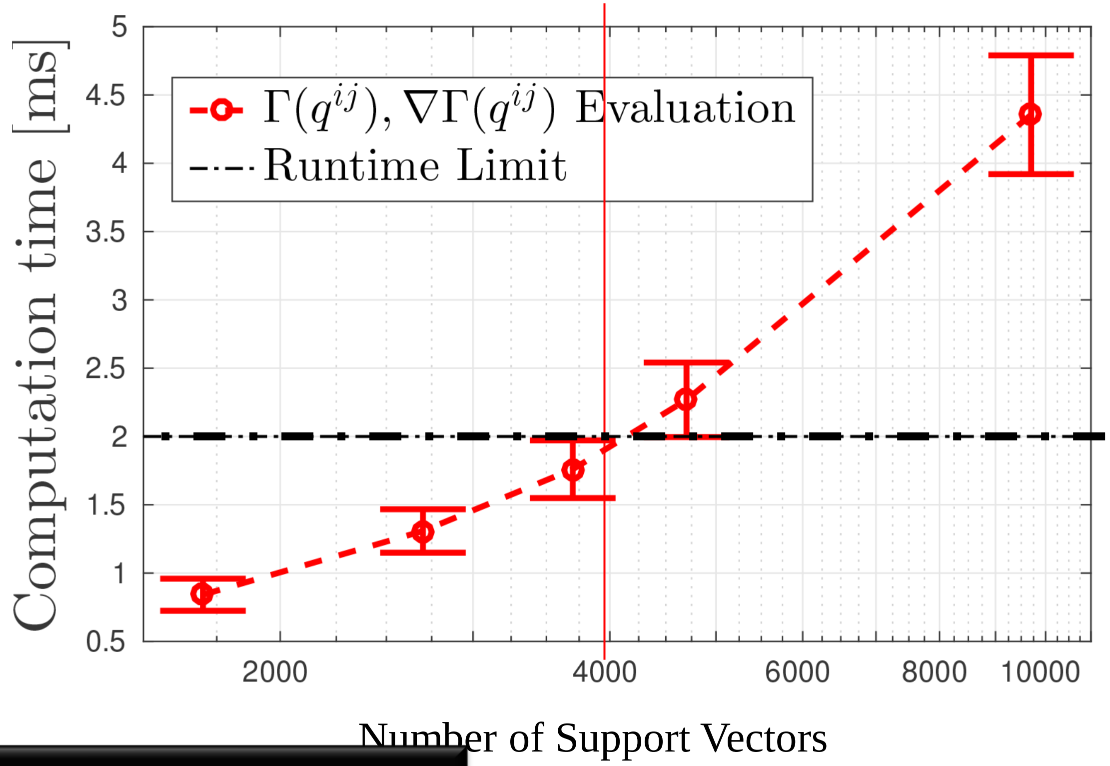


Sparse SVM

At testing time, computational costs grow ***linearly*** with number of Support Vectors.
We should not exceed 4K support vectors.
→ ***Sparse SVM.***

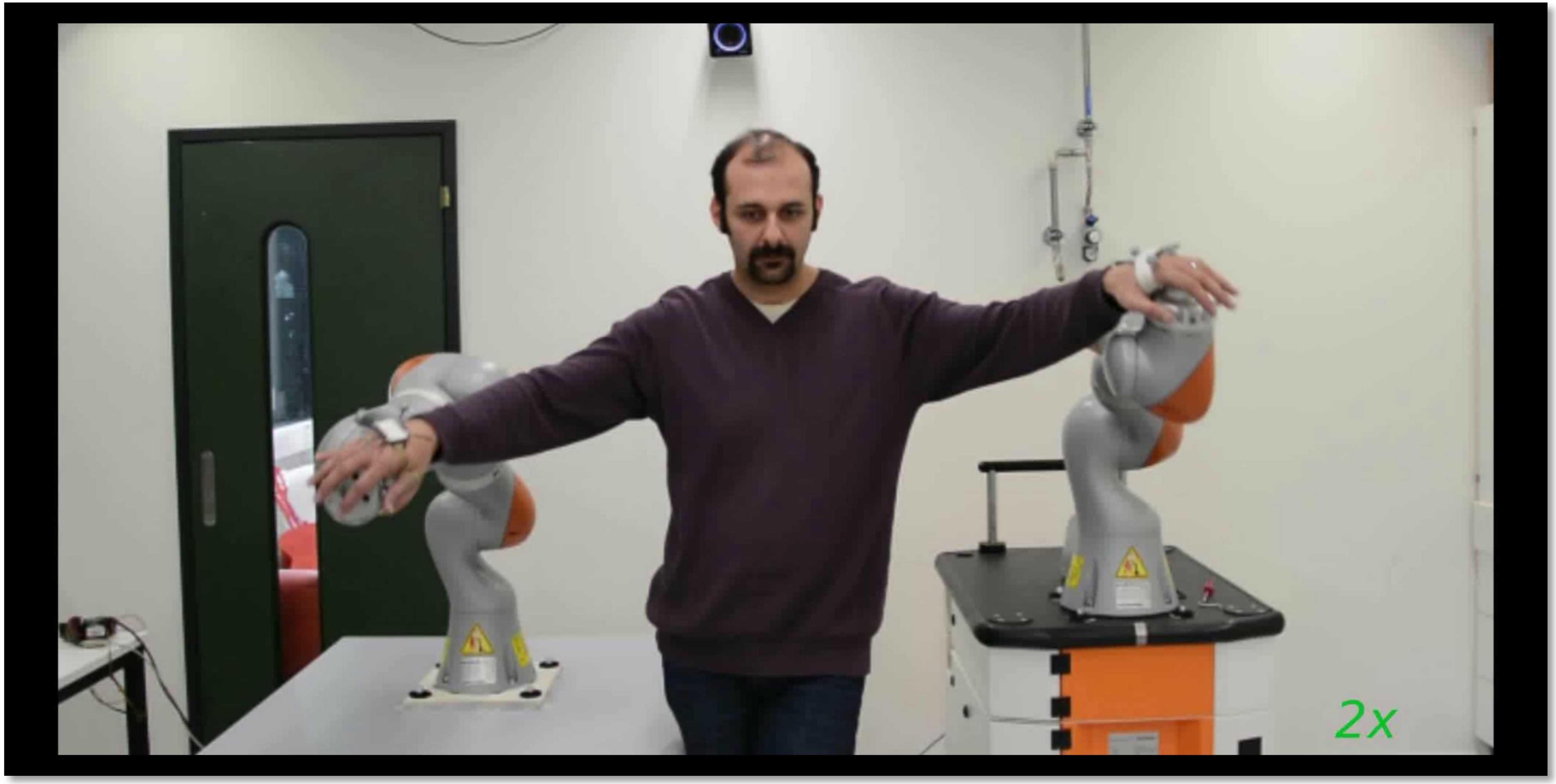
We have a budget of <4000 SVs ~





Sparse SVM results in very high (98%) accuracy with less than 2K SVs

Live Tracking of Moving Targets and Obstacle Avoidance



SUMMARY – 3rd Part

- To model *obstacles with very complex boundaries*, one can use non-linear classification approaches from machine learning.
  - One models the problem as a binary classification problem: feasible versus infeasible regions.

- Support vector classification (SVM) is well suited to learn such boundary, since, by construction of the decision function it learns a system of the form:

$$
\begin{array}{ll}
\Gamma(x) > 1 & \text{feasible states} \\
\Gamma(x) = 1 & \text{boundary} \\
\Gamma(x) < 1 & \text{infeasible states}
\end{array}
$$

- The boundary function of SVM is C1 (continuously differentiable) and hence one can compute the gradient along state space.
  - This can be used to force the flow to move away from the boundary and formulate this in a QP.

SUMMARY – 3rd Part

Limitations

- The approach is *no longer a closed-form DS*. It relies on the QP solver to find a feasible solution at run time.

- The boundary is only an approximation of the true boundary.

  - Unlike in the case where the obstacle's boundary was modelled through the convex hull, *we cannot ensure that the boundary covers perfectly the infeasible state space* and hence *we can no longer guarantee theoretically impenetrability of the obstacle.*

  - In practice, obstacle avoidance can be guaranteed if the learned model is learned with high accuracy and one takes a conservative approach, moving away from a safety margin to the boundary.