Adaptative parallelepipedic approximation of the image of a set by a nonlinear function

Maël Godard – Luc Jaulin – Damien Massé

SCAN 2025, Oldenburg, Germany

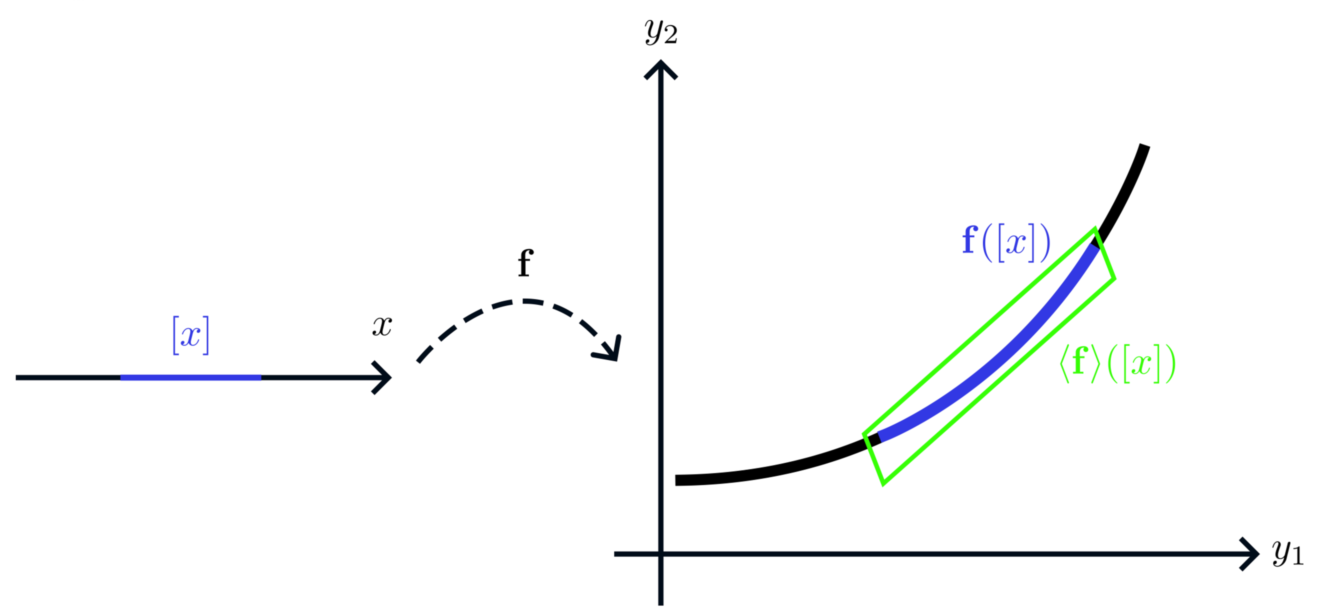
ENSTA INSTITUT POLYTECHNIQUE DE PARIS
Avec le soutien de Région Bretagne
Agence Innovation Défense
Lab-STICC
COLLEGES BRETAGNE LOIRE SCIENCES POUR L'INGENIEUR ET LE NUMERIQUE

Parallelepipedic Approximation

Let us consider a function $\mathbf{f}: \mathbb{R}^m \to \mathbb{R}^n$, $m \leq n$. A **parallelepiped inclusion function** of f is a function

$$\langle \mathbf{f} \rangle : \begin{array}{c} \mathbb{IR}^m \to \mathbb{PR}^n \\ [\mathbf{x}] \to \langle \mathbf{f} \rangle([\mathbf{x}]) \end{array}$$

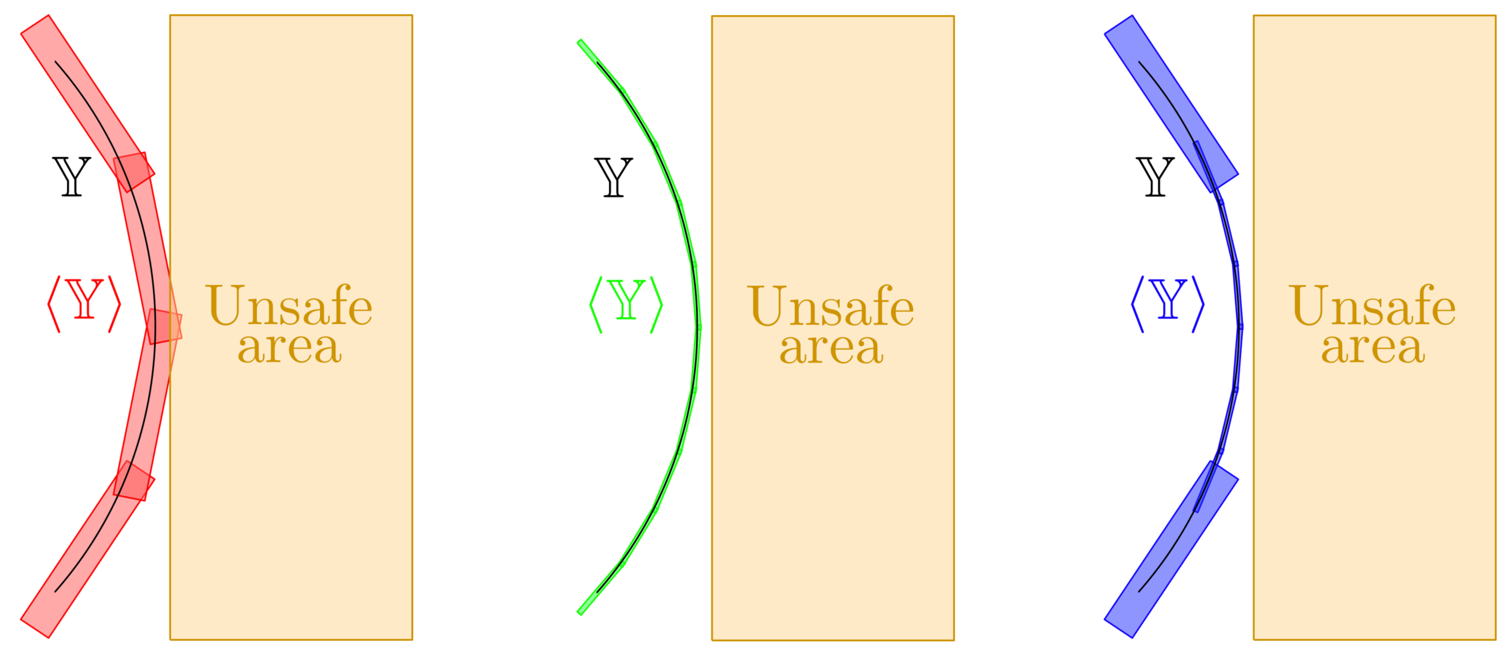
For a given box $[\mathbf{x}]$, this function returns a parallelepiped enclosing $\mathbf{f}([\mathbf{x}])$.



Parallelepipeds are efficient wrappers to enclose a set. They are a compromise between the **simplicity of boxes** and the **precision of zonotopes**.

Need for adaptivity

As any set based method, the use of parallelepipeds to enclose a set gives a **pessimistic** result. If the final objective is to assert that a constraint is satisfied, this pessimism can create an **ambiguity**.



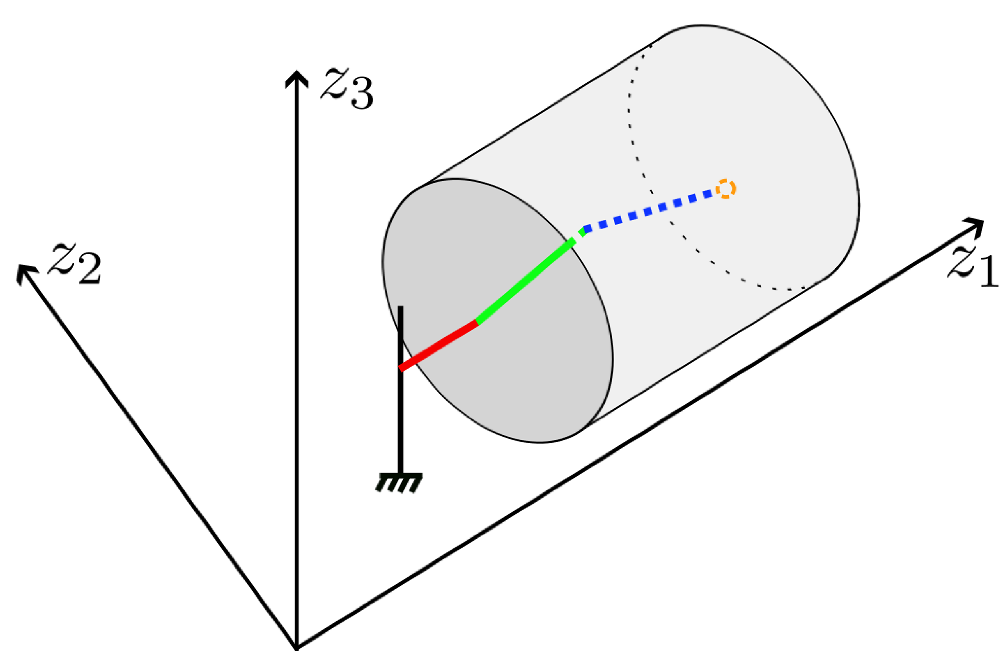
Ambiguity — No ambiguity — Adaptative

Using a high number of parallelepipeds everywhere is expensive time-wise and computational-wise. The idea presented here is then to use a low resolution everywhere, and to refine only the areas where it is needed. The result is an **adaptative parallelepipedic approximation**.

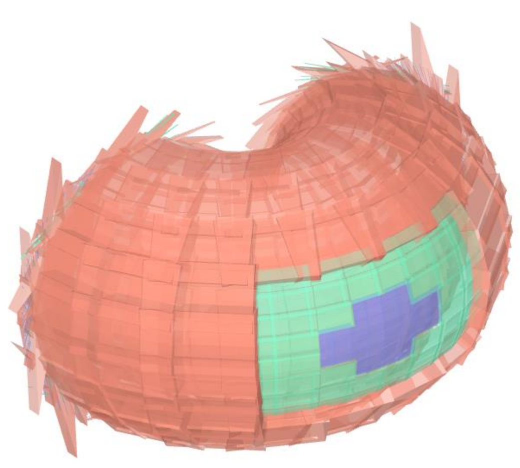
Use cases

Direct image computation

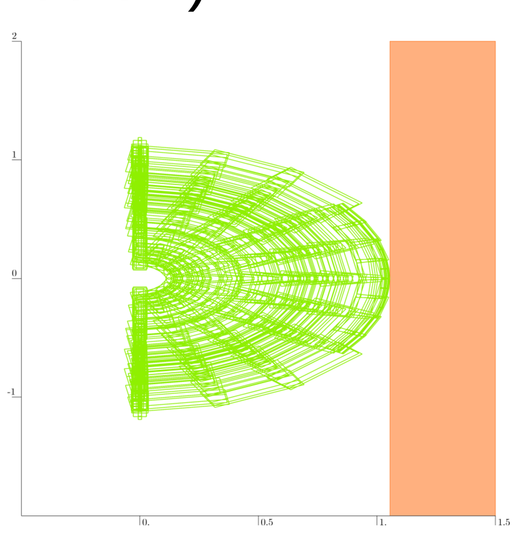
This method can be used to compute the **direct image of a set by a function**. For example, the computation of the workspace of a robotic arm.



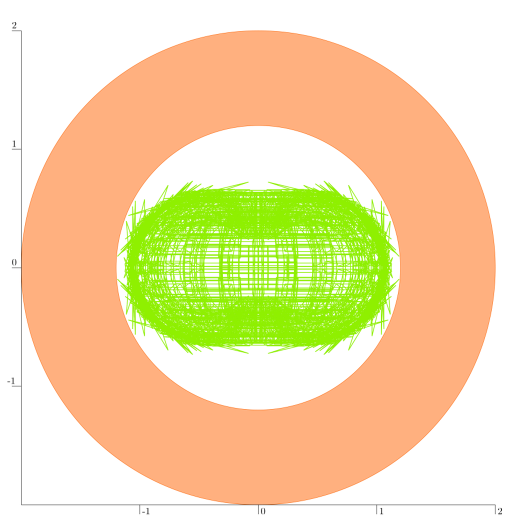
In this example we consider the set of the possible angles for each joint. The image set contains all the achievable positions for the effector. The constraint is that we want to **avoid collisions** between the effector and the grey cylinder (in red in both planes below).



3D workspace



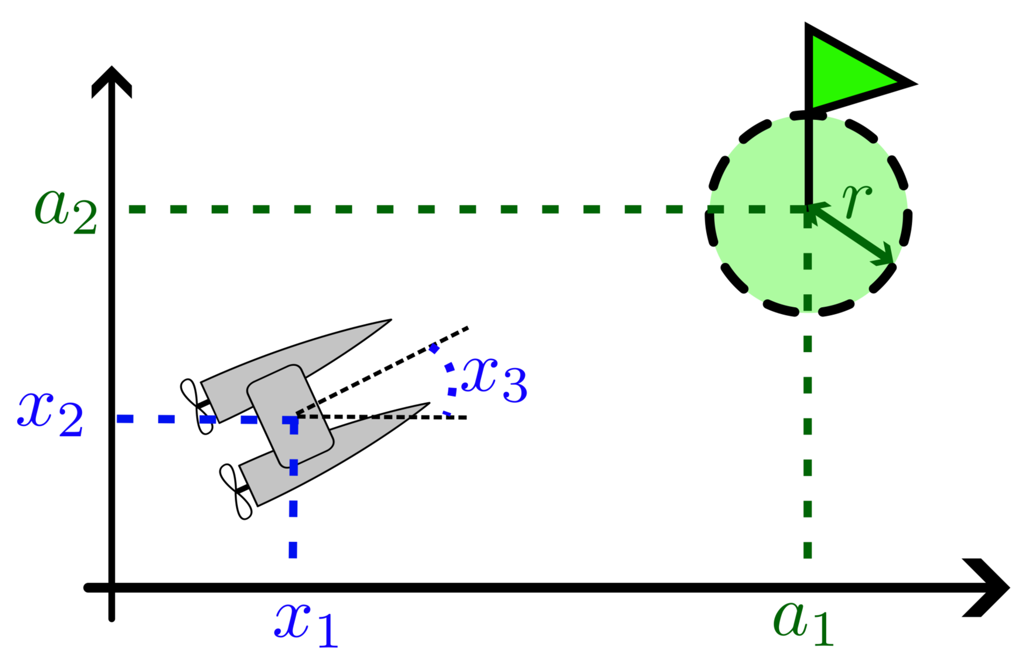
(z_1, z_2) plane



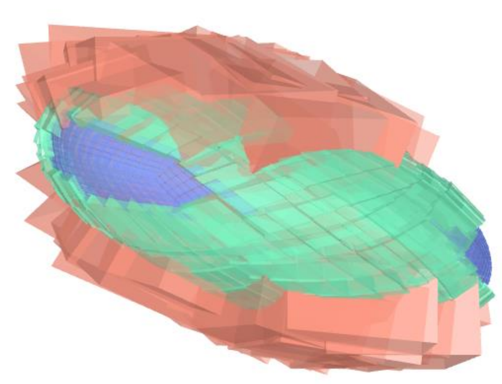
(z_2, z_3) plane

Reachable set computation

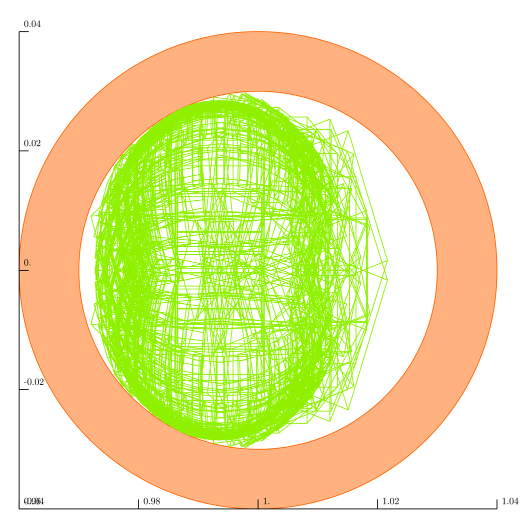
This method can be used to compute the **reachable set** of a dynamical system of the form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$.The state of the system is $\mathbf{x}$ and $\mathbf{u}$ is a **known input**.
As an example we will consider that the system is a robot defined by its position and its heading.



Here we consider the set of possible initial states for the robot. The image set contains all the possible states of the robot after a given time. We want to assert that the robot reaches the green area after a specific duration.



3D reachable set



(x_2, x_3) plane