# The Codac library

Simon Rohou[1], Benoît Desrochers[1]

[1] ENSTA Bretagne, Lab-STICC, UMR CNRS 6285, Brest, France
simon.rohou@ensta-bretagne.fr

**Keywords:** interval analysis, constraint programming, robotics

## Introduction

Codac (Catalog Of Domains And Contractors) is a C++/Python library providing interval tools for constraint programming over reals, trajectories and sets. In the field of robotics, complex problems such as non-linear state estimation, parameter estimation, delays, or SLAM can be solved in a very few steps by using constraint programming. Even though the Codac library is not meant to target only robotic problems, the design of its interface has been largely influenced by the needs of the above class of applications. Codac provides solutions to deal with these problems, that are usually hardly solvable by conventional methods such as particle approaches or Kalman filters.

## A framework of domains and contractors

Codac extends the tools proposed in the IBEX library to a wider class of problems. A *catalog* of domains such as intervals $[x]$, boxes $[\mathbf{x}]$, tubes $[x](t)$ (intervals of trajectories), thicksets $[\mathbb{X}]$ (intervals of sets) is available in Codac. These sets are contractible by *contractor* operators that aim at narrowing their bounds in a reliable way, according to several constraints defining a problem. The provided contractors are associated with publications from the literature and allow to deal with, for instance, non-linear constraints $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, inequalities, geometric constraints (distance, polar equation, circles), continuous differential

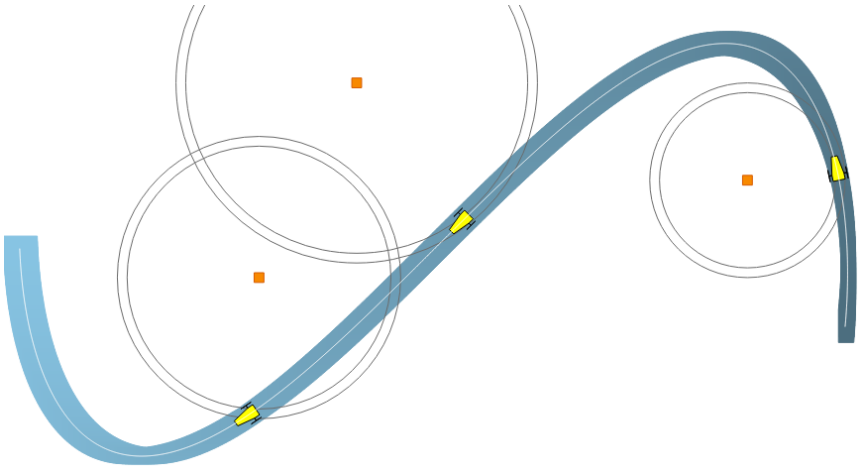

Figure 1: Guaranteed computation of a tube enclosing the feasible trajectories of a robot measuring bounded distances from three landmarks, without prior knowledge about its initial position.

equations: $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$, $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$, time uncertainties: $\mathbf{y} = \mathbf{x}(t)$ with $t \in [t]$, delays: $x(t) = y(t - \tau)$ [2], *etc.*

Domains and contractors can be combined in a *Contractor Network* that will manage the propagation of the contractions and ease the implementation of the related interval solver. In a few steps, we first: (1) define the initial domains (boxes, tubes) of our variables (vectors, trajectories) ; (2) take contractors from a catalog of already existing operators, provided in the library ; (3) add the contractors and domains to a Contractor Network ; (4) let the Contractor Network solve the problem ; (5) obtain a reliable set of feasible variables. The presentation will provide a simple application of Codac on a robotic problem.

## References


[1] S. Rohou, B. Desrochers and Others, *The Codac library, constraint-programming for robotics*, `http://codac.io`, 2022.

[2] R. Voges, Bounded-Error Visual-LiDAR Odometry on Mobile Robots Under Consideration of Spatiotemporal Uncertainties, *PhD Thesis*, 2020.