



Elaborato finale per il conseguimento  
della Laurea in Ingegneria meccanica

## **Synthesis of function generators cam-follower systems for rapid prototyping**

*Candidato: Federico Ciotti*  
*Matricola: 1593763*

*Relatore: prof. Nicola Pio Belfiore*  
*SSD: ING-IND/13*

**Abstract.** The aim of this project is the development of an open source software for the design of cam-follower mechanisms assigned the function of the follower travel or a set of heights to be linked with a given interpolation method. The software is addressed to research and rapid prototyping, thus is provided the option of exporting the result as collection of points in CSV form or directly as tridimensional STL model.

Lo scopo di questo progetto è lo sviluppo di un software open source per la progettazione di meccanismi a camma assegnata la funzione del movimento traslante della punteria ovvero un insieme di alzate da collegare tramite un'interpolazione assegnata. Il software è indirizzato alla ricerca ed alla prototipazione rapida, pertanto si è prevista la possibilità di esportare il risultato sotto forma di collezione di punti in formato CSV o direttamente come modello tridimensionale in formato STL.



## 1 Introduction

The program interface is an interactive console written in *Python 3* and it makes use of different libraries:

- *numpy* for array processing
- *matplotlib* for plotting
- *shapely* for geometric analysis
- *scipy* for interpolation
- *sympy* for expression parsing
- *pyclipper* for polygon offsetting
- *stl* for stl manipulation

## 2 Usage

Command parsing relies on *argparse* module, which takes charge of help generation, so every command's documentation may be requested with `-h` or `--help` option. Available commands are listed here:

```
usage: {help,exit,gen,update,load,save,draw,export,sim} ...
```

positional arguments:

```
{help,exit,gen,update,load,save,draw,export,sim}
  help          show this help message
  gen           generate, unspecified variables set to default
  update        update, unspecified variables unmodified
  load          load from file
  save          save to file
  draw          plot representation
  export        export stl model
  sim           dynamic simulation
  gen (and update), export and sim are discussed thoroughly in the following sections.
```

## 3 Follower travel generation

```
usage: gen travel [-h]
                  [--kind {spline,linear,harmonic,cycloidal,parabolic,polynomial}]
                  [--order ORDER] [-n N] [--steps STEPS] [--x0 X0] [--x1 X1]
                  (--input INPUT | --function FUNCTION)
```

optional arguments:

```
-h, --help          show this help message and exit
-k {spline,linear,harmonic,cycloidal,parabolic,polynomial}
                    kind of interpolation (default: linear)
--order ORDER, -o ORDER
                    spline/polynomial order (default: 3)
-n N                repetitions per cycle (default: 1)
--steps STEPS, -s STEPS
                    interpolation steps (default: 10000)
--x0 X0, -a X0      lower bound of function evaluation (default: 0)
--x1 X1, -b X1      upper bound of function evaluation (default: 1)
--input INPUT, -i INPUT
                    input file (default: None)
--function FUNCTION, -f FUNCTION
```



function of  $x$  (default: None)

The travel is generated from interpolation of points from an input file (list of semicolon-separated fraction of unit length and height) with a given method or as discretization of a function between given bounds. The command calls the method *gen* of a persistent instance of the class *Travel* which reads the input file or else parses the function, then it calls *update* which populates the class arrays  $x$  and  $y$  calling the appropriate method from the *interpolation* module or discretizing the function.

#### 4 Cam generation

```
usage: gen cam [-h] [--radius RADIUS] [--ccw] [--flat] [--offset OFFSET]
              [--fradius FRADIUS]
```

optional arguments:

```
-h, --help            show this help message and exit
--radius RADIUS, -r RADIUS
                        base radius (default: 0)
--ccw                  counterclockwise (default: False)
--flat, -f            flat follower (default: False)
--offset OFFSET, -d OFFSET
                        follower offset (default: 0)
--fradius FRADIUS, -q FRADIUS
                        follower radius (set 0 for knife edge) (default: 0)
```

a

#### 5 STL exportation

#### 6 Simulation

#### 7 Utilities

```
parser_update_cam.add_argument('--nonflat', dest='flat', action='store_const', const=1)
```

```
\input ../main.py
```

In [ ]: