

iLQRSolver.py is a library used to compute a sequence of control to reach a specified point with a specified system and cost function.

In order to use the solver, you have to implement a class which describe de dynamic model of the system and another which describe the cost function you will use to solve your problem.

## dynamic model

This class is the one which describe the system you want to command. It needs to have particular attributes and methods.

### attributes

Two attributes are mandatory for the solver to work.

```
self.stateNumber  
self.commandNumber
```

stateNumber attribute allow the solver to know how many variable your state contains.

commandNumber specified the number of commands your system have.

These 2 attributes are supposed to be unsigned integer.

Both these attributes make the Solver as generic s possible.

### methods

Two methods are mandatory for the solver to work.

```
def computeAllCostDeriv(self,extra arguments ...):  
def computeFinalCostDeriv(self,extra arguments ...):
```

The solver class need to be instantiated with the dynamic model of the system and the cost function you want to apply.

```
solver = ILQRSolver(model,costFunction)
```