ECOLE CENTRALE DE NANTES

MASTER CORO-IMARO

Humanoid Robotics

Design a cyclic reference trajectory by optimization

Serena Roncagliolo, Camille Vindolet, Josep Rueda Collell

28 December 2020

Contents

1 Introduction

The purpose of this Lab is to design a periodic walking of duration T. In order to solve this problem we will follow these steps:

1. Write a main program where we can define the physical parameters, the initial set of optimization variables and the instructions to call *fmincon* matlab function. We will then plot the interesting variables
2. Write the program *optimwalk* as a Matlab function with the optimization variables. In this function we will consider the impact model to deduce q^+
3. consider the reduced dynamic inverse model of the compass robot in single support and deduce the ground reaction **R**
4. define the criterion
$$C = \frac{1}{L_{step}} \sum_{k=1}^{N=100} \Gamma^2(k\Delta)\Delta \quad (1)$$
and we evaluate it using sampling period $\Delta = T/100$
5. define the chosen constraints and find the optimization variables using *fmincon*, giving the profiles of q_i, $\dot{q}_i$, R_x, R_y, Γ_i

2 Design optimized reference trajectory

2.1 Main Program

In the main program we start by defining the physical parameters that define our walking robot and the initial state for the motion.
In particular we may test our code in two situations: We simulate our final code using the following parameters:

- l=0.8m
- m=2 kg
- I=0.1 kg·m²
- s=0.5m
- θ=3*pi/180 rad
- $q_1 = -0.1860rd$
- $\dot{q}_1 = -2.0504rd/s$
- $\dot{q}_2 = -0.0428rd/s$

where l is length of the leg, m the value of the mass of one leg, I the inertia, s is the distance between the position of center of gravity and position of the hip.
Then it is necessary to define the optimisation variables. We first decided to define $Jsolcons0 = [q_1, \dot{q}_1, \dot{q}_2]$ as q2 is obtained by the double support configuration equation $q_2 = \pi - 2q_1$. We first set the time duration of a step to be T and the intermediate position where the legs are aligned to happen at time 0.5T. Then we chose to optimise

the time duration of a step and the intermediate time.
The optimisation is solved by the function *fmincon*. This function, tries to solve a system, trying to minimise a particular function. In our case, we want our robot to perform a particular motion. We are trying to optimise it in such a way that the needed torques to perform the motion, are the smallest possible.
The function **fmincon** is a solver for non linear programming. Its instructions are defined as follows:

```
[x,fval,exitflag,output]=fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)
```

It finds the minimum of a problem specified by

$$\min_x f(x) \text{ such that } \begin{cases} c(x) \leq 0 \\ c_{eq}(x) = 0 \\ A\boldsymbol{x} \leq \boldsymbol{b} \\ A_{eq}\boldsymbol{x} = \boldsymbol{b_{eq}} \\ lb \leq \boldsymbol{x} \leq ub \end{cases} \tag{2}$$

where $\boldsymbol{b}$ and $\boldsymbol{b}_{\text{eq}}$ are vectors, A and A_{eq} are matrices, $c(x)$ and $c_{\text{eq}}(x)$ are functions that return vectors, and $f(x)$ is a function that returns a scalar. $f(x)$, $c(x)$, and $c_{\text{eq}}(x)$ can be nonlinear functions. **x**, lb, and ub can be passed as vectors or matrices. In our case we call:

```
[Jsolcons,Fval,EXITFLAG] =
fmincon("resol",Jsolcons0,[],[],[],[],lb,ub,"constraint",options)
```

Resol is the function to optimize and *Jsolcons0* are the optimization variables [q1 q1d q2d]. A = [], b=[], A_{eq}=[] and b_{eq}=[] because the function is not subject to linear equalities and inequalities. lb and ub are the lower and upper bounds. We select them so that the solution is physically possible. We limit the angles to be between $\frac{\pi}{2}$ and $-\frac{\pi}{2}$ and the velocities to be under 50rad/s. In case we decide to optimise the time of intermediate position we choose it to be between 0 second and the time of the step. In case we optimise the time of a step we choose it to be between 0 second and 5 seconds. This is an arbitrary choice that can be modified depending on the application. The optimisation variables will be such that the energy is minimised.

```
lb = [-pi/2 -2*pi -10 -10]
ub = [+pi/2 +2*pi +10 +10]

options = optimset('Display','iter','MaxFunEvals', ? ?,'MaxIter', ?
?,'LargeScale','off')

[Jsolcons,Fval,EXITFLAG] =
fmincon("resol",Jsolcons0,[],[],[],[],lb,ub,"constraint",options)
```

2.2 Resol function

We write the program *resol.m* as a Matlab function which takes the optimisation variables as its inputs.

At the beginning of this function we take into account the impact model so that we can deduce $\dot{q}^+$:

$$A_1((T))(\dot{q}^+ - \dot{q}^-) = J_2^T I_2 \tag{3}$$

$$J_{2R}(\dot{q}^+ - \dot{q}^-) = 0 \tag{4}$$

We can compute A_1 and $J_{2\mathrm{R}}$ using the function *function_impact* of the first lab obtaining :

$$A_1 = \begin{bmatrix} 2m & 0 & sm(cosq_1 - cos(q_1+q_2)) & -mscos(q_1+q_2) \\ 0 & 2m & ms(sinq_1 - sin(q_1+q_2)) & -mssin(q_1+q_2) \\ sm(cosq_1 - cos(q_1+q_2)) & ms(sinq_1 - sin(q_1+q_2)) & 2ms^2+2I & I+ms^2 \\ -mscos(q_1+q_2) & -mssin(q_1+q_2) & I+ms^2 & ms^2+I \end{bmatrix} \tag{5}$$

$$J_{2r} = \begin{bmatrix} -lcos(q_1+q_2) & -lcos(q_1+q_2) & 1 & 0 \\ -lsin(q_1+q_2) & -lsin(q_1+q_2) & 0 & 1 \end{bmatrix} \tag{6}$$

$\dot{q}^+$ will then be deduced as:

$$\dot{q}^+ = (A_1((T)))^{\text{-}1}(J_2^T I_2) + \dot{q}^{\text{-}} \tag{7}$$

Inside the function *resol* we also define the polynomial functions, by considering coefficients connected to:

- initial configuration
- intermediate configuration
- final configuration

The aim of this section is to define a trajectory using polynomial functions:

The trajectories followed by $\mathbf{q_a} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$, $\dot{\mathbf{q}}_\mathbf{a} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}$ are of the form:

$$q_1 = a_0 + a_1\mathbf{t} + a_2\mathbf{t}^2 + a_3\mathbf{t}^3 + a_4\mathbf{t}^4 \text{ and } q_2 = b_0 + b_1\mathbf{t} + b_2\mathbf{t}^2 + b_3\mathbf{t}^3 + b_4\mathbf{t}^4 \tag{8}$$

$$\dot{q}_1 = a_1 + 2a_2\mathbf{t} + 3a_3\mathbf{t}^2 + 4a_4\mathbf{t}^3 \text{ and } \dot{q}_2 = b_1 + 2b_2\mathbf{t} + 3b_3\mathbf{t}^2 + 4b_4\mathbf{t}^3 \tag{9}$$

We are stacking the parameters a_i and b_i with i going from 0 to 4 in a vector $\mathbf{A}$ and $\mathbf{B}$. For a simple trajectory we are deciding the initial and final angular positions and velocities and the position of the intermediate angular positions, $\mathbf{q}_{a0}, \dot{\mathbf{q}}_{a0}, \mathbf{q}_{af}, \dot{\mathbf{q}}_{af}, \mathbf{q}_{am}$ and the time duration of the trajectory that we call T and that is equal to 1 second. We can then compute A to deduce the trajectories for $\mathbf{q}_a$ and $\dot{\mathbf{q}}_a$. In matrix form we obtain :

$$\begin{bmatrix} q_{10} \\ \dot{q}_{10} \\ q_{1f} \\ \dot{q}_{1f} \\ q_{1m} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 \\ 1 & t_m & t_m^2 & t_m^3 & t_m^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} \tag{10}$$

Also written:

$$\bar{\mathbf{q_1}} = \mathbf{TA} \tag{11}$$

and:

$$\begin{bmatrix} q_{20} \\ \dot{q_{20}} \\ q_{2f} \\ \dot{q_{2f}} \\ q_{2m} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 \\ 1 & t_m & t_m^2 & t_m^3 & t_m^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} \tag{12}$$

Also written:

$$\bar{\mathbf{q_2}} = \mathbf{TA} \tag{13}$$

Thus we can compute $\mathbf{A}$ and $\mathbf{B}$ to deduce the trajectories for $\mathbf{q}_a$,$\dot{\mathbf{q}}_a$:

$$\mathbf{A} = \mathbf{T}^{-1}\bar{\mathbf{q_1}} \text{ and } \mathbf{B} = \mathbf{T}^{-1}\bar{\mathbf{q_2}} \tag{14}$$

Once we know $\mathbf{A}$ and $\mathbf{B}$ we obtain the trajectories $\mathbf{q}_a$,$\dot{\mathbf{q_a}}$. All that is left to do is to decide which values of $\mathbf{q_{a0}}, \mathbf{q_{af}}, \dot{\mathbf{q_{af}}}, \mathbf{q_{am}}$ we want.
For this we can base ourselves on the trajectory followed by the compass robot in the unstable motion analysed in the lab 2 of HUMRO.
We decide the initial, intermediate and final conditions knowing that the initial state is obtained after an impact and that the final one is obtained before an impact. There is a symmetry between the two. We first define the final state before impact, because of the way we defined the problem the final state has to be optimised:

- q_{1f} = Jsolcons(1)
- q_{2f} = pi - 2q_{1f} which is the condition for double support
- $\dot{q_{1f}}$ = Jsolcons(2)
- $\dot{q_{2f}}$ = Jsolcons(3)

We then compute the impact to obtain the initial condition, just after impact. The first and final step show the following symmetry:

- $q_{10} = -q_{1f}$
- $q_{20} = 2\pi - q_{2f}$

The condition for the intermediate angular positions is:

- $q_{1m} = 0$
- $q_{2m} = \pi$

We consider the reduced dynamic inverse model of the compass robot in single support given by:

$$A(\boldsymbol{q}) + H(\boldsymbol{q}, \dot{\boldsymbol{q}}) + G(\boldsymbol{q}) = \Gamma \tag{15}$$

To compute the inverse model we use the function *function_dyn* from first lab:

```
[A,H,Hg] = function_dyn(q1(n),q2(n),q1d(n),q2d(n));
Gamma(:,n) = A*[q1dd(n);q2dd(n)]+(H+Hg);
```

We can then determine the ground reaction **R** using the function *function_reactionforce.m* from Lab 1:

```
function_reactionforce(q1(n),q2(n),q1d(n),q2d(n),q1dd(n),q2dd(n));
```

We then define the criterion:

$$C = \frac{1}{L_{step}} \int_0^T \Gamma^T \Gamma dt \tag{16}$$

Remember that this function will try to find the smallest value of Criterion without violating the constraints. We evaluate this integral using the backward difference method using sampling period $\Delta = T/100$:

$$C = \frac{1}{L_{step}} \sum_{k=1}^{N=100} \Gamma^2(k\Delta)\Delta \tag{17}$$

In Matlab:

```
C = sum(1/Lstep*Gamma(1,n)^2*delta); sum(1/Lstep*Gamma(2,n)^2*delta);
```

The choice of the constraints is given by

$$\begin{cases} L_{step} > 0.2m \\ R_y > 0 \\ I_y > 0 \\ \frac{R_x}{R_y} < 0.7 \\ \frac{I_x}{I_y} < 0.7 \\ |\dot{q}_1| < 3rd/s \\ |\Gamma_i| < 50Nm \quad where \quad i = 1, 2 \end{cases} \tag{18}$$

3 Conclusions

Finally, we end up by plotting the kinematic variables, as long as the reaction forces at the ground and the torque of each articulation.

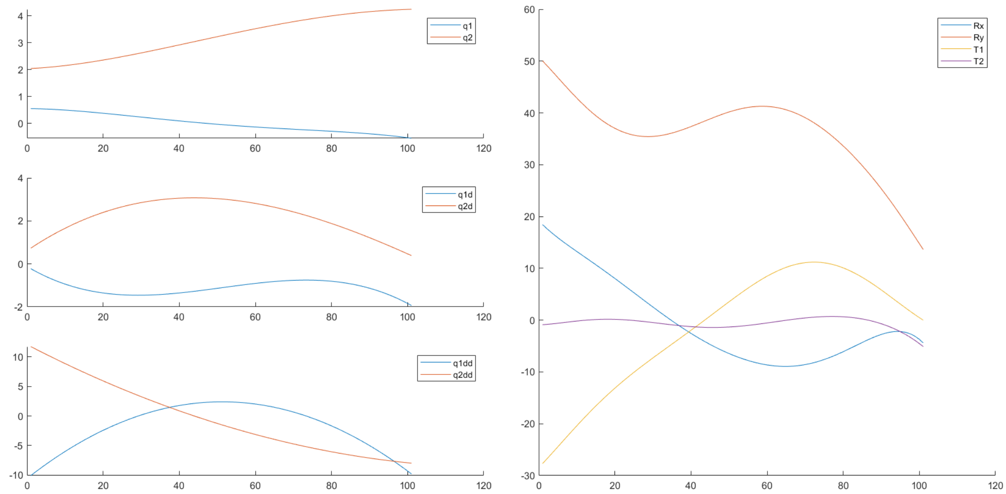

Figure 1: Plot

We can graphically see that the torques are never bigger than 50 N·m and the reaction force in Y is always positive. As such we can see that the constraints are met.