Optimal Kinematic Design of Robots Lab 1 report

Fabio Conti, Francesco Pagano

September 2022

1 Introduction

The aim of this lab is to show how is it possible to use the workspace in the Cartesian space of a planar SCARA robot as a design tool of the robot itself. Starting from the workspace shape it is possible to optimize the link length of the robot for a specific task so that we can adapt the robot to the environment. Our work plan was to develop a Matlab script capable of plotting the workspace boundaries of a planar RR SCARA robot by taking into account a **disc-shaped obstacle** in the way of only the first link and **Joint limits**.

Furthermore, we had to determine the robot's best link length ratio in a *point-to-point* task cutting process task.
The code we developed is composed of the following scripts:
Lab1.m, *find_angle.m*, *plot_circle*, *plot_bound.m*.

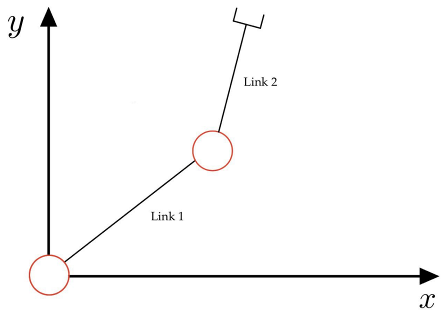


Figure 1: Schematic of the planar SCARA robot

2 Methods

We approached the problem by dividing the final task into some easier ones in order to add up code gradually and easily meet the lab's requirements. For achieving the final robot workspace plot, we went through the following steps:

1. Plot the workspace without neither joint limits nor obstacles;
2. Add joint limits;
3. Add the obstacle.

Afterwards, by using the provided interactive user interface, we managed to determine the optimal link length ratio for the aforementioned tasks.

2.1 Workspace without neither joint limits nor obstacles

The DGM (Direct Geometric Model) of the robot is:

$$x = l_1 cos(q_1) + l_2 cos(q_1 + q_2)$$

$$y = l_1 sin(q_1) + l_2 sin(q_1 + q_2)$$

By exploiting these equations we could easily plot the inner and outer bound of the workspace which also represent the singularity lines of the manipulator. The inner bound is given by setting the second joint angle equal to zero ($q_2 = 0$) and plotting the x, y position of the end effector. Instead, the outer bound is given by setting $q_2 = \pi$. These two position represent the **singularity conditions** for the robot in this configuration. In both cases, since we're not considering any joint limit, q_1 will not be bounded varying from 0 to 2π.

Here's the plot we obtained:

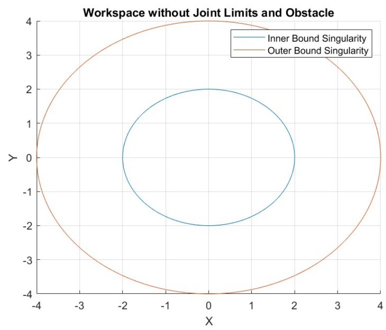
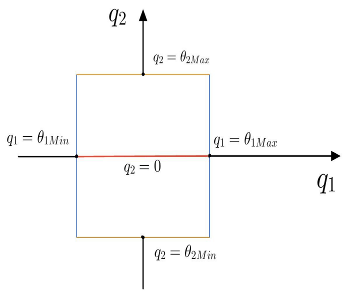


Figure 2: Workspace without joint limits and obstacle

2.2 Adding Joint Limits

To build the *reduced workspace* in the Cartesian space we started imposing the joint limits in the **joint space**. Our SCARA robot joint space is a rectangle and, for each side of it, we plotted the x, y end effector coordinate given by the *DGM* of the robot. the two vertical sides of the joint space can be translated into Cartesian space by plotting the *DGM* by varying the angle of the second joint between the minimum value and the maximum value of the limit while keeping the angle of the first joint fixed (once at the maximum limit and once at the minimum limit). Likewise, the two horizontal sides of the joint space can be converted to the Cartesian space by plotting the *x, y* positions given by the *DGM* by varying the angle of the first joint between the minimum value and the maximum value of the user-set limits while keeping the second joint angle fixed once at the maximum value and once at the minimum value. Finally, to obtain the outer limit of the workspace we had to plot the x, y positions by varying q_1 and setting $q_2 = 0$. Here's the plot we obtained:

(a) Workspace with joint limits



(b) Workspace in the Joint Space with joint limits

2.3 Adding the Obstacle

Following the line of what we had already done with the previous step, we started thinking about the new bounds and movement limitations inside the joint space when adding the disc obstacle into the scenario. Two extra vertical lines limits were added in the span of the first joint to discard the blocked joint space portion. To compute the two new joint limits we created a dedicated function named *"find_angle.m"*. Such function calculates the angles within which the robot is obstructed starting from the user-defined dimension and position of the obstacle in space. First of all, the angle between the horizontal line and the circle centre is computed as follows:

$$\alpha = tan^{-1}(\frac{x_c}{y_c})$$

where x_c, y_c are the centre coordinate of the circle. Afterwards, we found the angle span to make the robot avoid the disc obstacle as angle between the previously found vector and the perimeter of the obstacle:

$$\beta = sin^{-1}(\frac{r}{l})$$

with:

- r as the circle radius and,
- l as the distance between the circle centre and the origin of the base frame. ($l = \sqrt{x_c^2 + y_c^2}$)

The two new joints limits were obtained as:

$$\theta_{max,obstacle} = \alpha + \beta$$

$$\theta_{min,obstacle} = \alpha - \beta$$

Here's a drawing to clarify our computations:

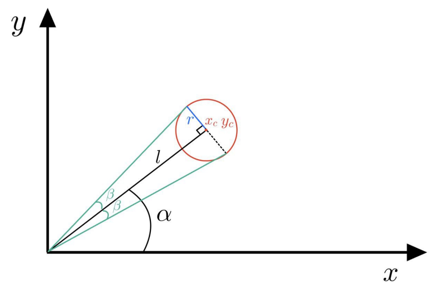
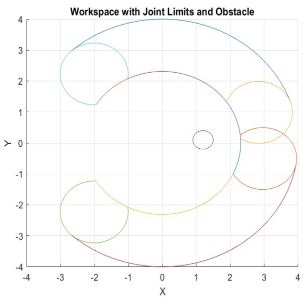
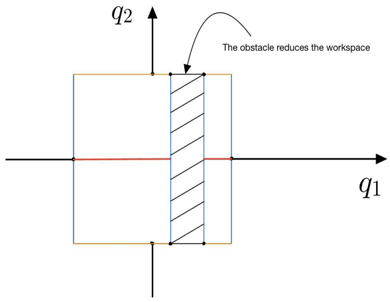


Figure 4: Schematic for the computation of the new joint limits due the obstacle

$\theta_{max,obstacle}$ and $\theta_{min,obstacle}$ are the new joint limits due to the presence of the obstacle. Therefore, the procedure for computing the resulting workspace is exactly the same as before but in this case two more limits will be taken into account, obtaining:



(a) Workspace with joint limits and Obstacle



(b) Workspace in the Joint Space with joint limits and Obstacle

In these cases the robot parameter have been set offline as follows:

- *Joint 1*: $\theta_{1,min} = -132°$, $\theta_{1,max} = 132°$
- *Joint 2*: $\theta_{2,min} = -141°$, $\theta_{2,max} = 141°$
- *Link 1 and 2*: $l_1 = 3m$, $l_2 = 1m$
- *Obstacle position and radius*: $(x_c = 2m, y_c = 2m); r = 0.5m$

2.3.1 Results

The plots we obtained by gradually adding constraints show how the robot's work-envelop changes shape and in particular how it shrinks down. Fig 5a is the final result and we can see how the workspace is composed of two areas that are not connected to each other because of the obstacle. Moreover, the robot is not able to make motions at 360 degrees but it can work only in the regions in front of it due to limits of the joints.

3 Determine the optimal link length ratio

The second part of the lab involved the use of a provided *MatLab interface* made specifically for robot design. The main propose of this program is to draw the envelope of the workspace of a generic RR SCARA robot in the Cartesian and joint space while considering multiple design parameters and the presence of an obstacle inside the work space.
Considering the following design constraints:

- *Joint 1*: $\theta_{1,min} = -132°$, $\theta_{1,max} = 132°$
- *Joint 2*: $\theta_{2,min} = -141°$, $\theta_{2,max} = 141°$
- *Maximum reach*: $l_1 + l_2 = 2m$
- *Obstacle position and radius*: $(x_c = 1.2m, y_c = 0.1m); r = 0.3m$

The aim of this portion of the lab was to make use of the interface to design the best link length ratio possible for the SCARA robot for each of the following manufacturing tasks:

1. *point-to-point tasks*: Picking parts from an L-shaped palette of dimensions $2m \times 1m$ and of width $0.5m$ and place them into another identical palette.

2. *process tasks*: cutting an L-shaped plate of dimensions of dimensions $2m \times 1m$ and of width $0.5m$.

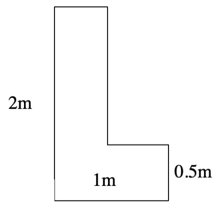


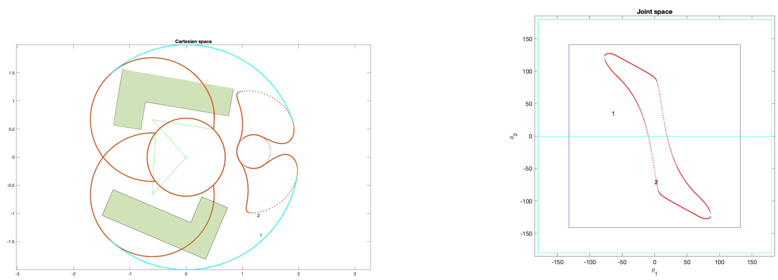
Figure 6: L-shaped plate used in the two task applications

3.1 Point-to-point tasks

SCARA robots are typically used to quickly pick up and move relatively light objects across short distances. In this case study, the designer's goal is to specify the **appropriate link length** required to contain both the L shaped palettes and making sure that the regions where these two shapes lay are connected.
The first thing we did was to **maximize the working area** of the robot by tweaking the links' length to include the two L-shape areas. This process led us to define the lengths as follows:

- $l_1 = 0.9m$, $l_2 = 1.1m$

These parameters combined with the fixed ones led us to the following *work-envelop* graphs:



(a) Workspace in Cartesian space with joint limits and Obstacle

(b) Workspace in the Joint Space with joint limits and Obstacle

Where for picture 7a: *Green line*: represent the robot links at the joint limit configuration. *Thick red line*: represent the workspace limits due to joint limits. *Thin red line*: represent the obstacle representation. *Blue line*: represent the workspace limits due to singularities. *Green areas*: represent the palettes for the picking up task.
And for picture 7b: *Blue line*: represent the workspace limits due to joint limits. *Red line*: represent the obstacle representation. *Light blue line*: represent the workspace limits due to singularities. *Numbers*: represent the configurations the robot can deal with: either elbow up (1) or elbow down (2).

3.1.1 Comments & Results

The Fig 7a clearly shows how the green palettes are completely included inside the work envelope of the robot. This configuration allows the robot to reach any point of the surfaces with the

end-effector for the pick-up task. The Fig 7b shows how the movement limitation caused by the the joint limits and the obstacle do not prevent the robot to start from any point belonging to the first palette and reach another point on the second one. Therefore, we can say that the two palettes are part of 2 n-connected regions. This means that from any starting position located on the area of the first palette, the second one can be always reached.
Thanks to the provided interface, it is possible to plan a trajectory inside the joint and Cartesian space graphs. If a trajectory simulation is carried out throughout of the graphs we can clearly see how the the robot is capable of reaching the two areas seamlessly. From the simulation we can notice that every time the trajectory in the joint space crosses *the Light Blue* line the robotic arm changes aspect (from elbow up to down) passing through a singular configuration. Though, this behaviour doesn't prevent the task to be fulfilled.

3.2 Process tasks

The task of last point of the lab concerns the cutting of a sheet of the same dimensions as the palettes. Similar to welding, milling, grinding, and many other processes, cutting tasks are related to the robot's ability to **move along predetermined lines**. A requirement for this category of tasks is that the EE must not leave the trajectory during the execution of the task. The only condition for the given path (which corresponds to the perimeter of the L-shaped palette) to be continuously tracked is that the robot does not have to change its posture during motion; this means that the path must be part of a t-connected region inside the workspace and therefore reachable in one single aspect.

Since we already had a fully covered L-shaped area from the previous design we decided to try and keep the old design for the current task:

- $l_1 = 0.9m$, $l_2 = 1.1m$

As we can see from the following graph:

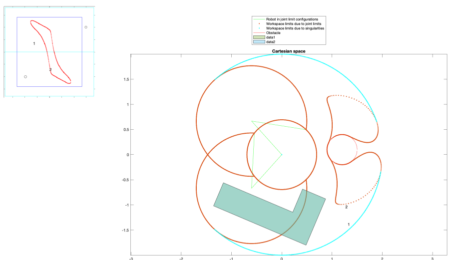


Figure 8: Cartesian and joint space representation for the cutting task + L-area

The work envelop completely covers the L-shaped area without ever leaving the current aspect which was the objective of this design.

3.2.1 Comments & Results

Thanks to the simulation tool, we can clearly see that the perimeter of the green area is perfectly scored in the Cartesian space. In the chosen configuration the SCARA robot can generate linear trajectories without leaving the established path and without changing aspect configuration leading to a perfect cut and the fulfilment of the task.