NTNU

Student Project 2

Group members:

- 755110
- 759135
- 759144

Abstract

The theory related to a specific instance of the inverse kinematic problem in two dimensions is investigated. A cost function penalizing large changes in joint angles is minimized under the restriction of visiting points along a given path. An augmented Lagrangian method in combination with BFGS is chosen for this purpose, and is shown to be successful. A barrier method developed in order to minimize the same problem with additional angular freedom restrictions is unfortunately not successful in the general case.

Introduction

This paper investigates the properties relating to an inverse kinematic problem. Specifically, we consider the problem of moving a robot arm consisting of n joints connected by n segments of different lengths. The last segment is connected to an end-effector. We are interested in minimizing the cost of moving the end-effector through a sequence of points. This can be expressed as an optimization problem with a set of equality constraints, and can then solved by the use of a suitable numerical optimization method. It can be argued that the augmented Lagrangian method, using the quasi-Newton method BFGS for solving the unconstrained sub-problems, is well suited.

Furthermore, the freedom of rotation of the joints is restricted, introducing inequality constraints to the optimization problem.

In the following sections the theory is dealt with more closely and potential challenges are discussed. Numerical methods are then presented and applied in an attempt to solve the problems subject to the different constraints. The results, including convergence analysis and a numerical investigation of the configuration space, is then presented.

Theory

The robot arm in question consists of n segments of length l_i going from joint i to joint $(i+1)$ for $i = 1, 2, \ldots, n-1$. The last segment goes from joint n to an end-effector. The angle between the ith segment and the $(i-1)$th segment is denoted by ϑ_i. The position of the first joint is chosen to be the origin, and we consider ϑ_1 to be the angle between the first segment and the x-axis.

The joint space, the set of all possible joint configurations, is denoted by $\mathscr{J}$. The joint space is equal to $\mathbb{R}^n$ if we don't impose any restrictions on the values of ϑ_i. This is considered to be the case until otherwise stated. The robot arm also has an associated configuration space, denoted by $\mathscr{C}$, which consists of all the points in $\mathbb{R}^2$ that can be reached by some configuration of the joints. Finding the position in the configuration space that corresponds to a position in the joint space can be expressed as a function $F : \mathscr{J} \to \mathscr{C}$. The function is

$$F(\vartheta) = \sum_{i=1}^{n} l_i \left(\cos(\sum_{j=1}^{i} \vartheta_i) \quad \sin(\sum_{j=1}^{i} \vartheta_i) \right)^T . \tag{1}$$

The index of the longest segment is denoted m. The maximum reach of the end-effector is equal to the sum of all segment lengths $\sum_{i=1}^{n} l_i$. If the longest segment is longer than the sum of the remaining segments then there is a circular region around the origin that becomes unreachable. In such a case the configuration space is an annulus centered at the origin, otherwise it is a disk.

The set of points satisfying these restrictions can be summarized by

$$\mathscr{C} := \{ x \in \mathbb{R}^2 : l_m - \sum_{i=1, i\neq m}^{n} l_i \leq ||x|| \leq \sum_{i=1}^{n} l_i \} . \tag{2}$$

The problem of interest is to find a path for the robot arm such that the end-effector visits a set of given points $p^{(1)}, \ldots, p^{(s)} \in \mathscr{C}$ and then returns to $p^{(1)}$. This should be done such as to minimize the total rotation of the joints while the arm traverses the path. One way of doing this is formulated in [2]: Let $\vartheta^{(1)}, \ldots, \vartheta^{(s)}$ denote the joint angles when the end-effector is at the points $p^{(1)}, \ldots, p^{(s)}$ respectively. Minimize

$$E(\Theta) := \frac{1}{2} \left[\left\| \vartheta^{(2)} - \vartheta^{(1)} \right\|^2 + \cdots + \left\| \vartheta^{(s)} - \vartheta^{(s-1)} \right\|^2 + \left\| \vartheta^{(1)} - \vartheta^{(s)} \right\|^2 \right]$$

$$\text{subject to} \quad F(\vartheta^{(j)}) = p^{(j)} \text{ for } j = 1, \ldots, s, \tag{3}$$

where $\Theta := (\vartheta^{(1)}, \ldots, \vartheta^{(s)}) \in \mathscr{J}^s$. Here and throughout this paper $||\cdot||$ denotes the l_2-norm.

Let $\Delta_i^{(j)} = \vartheta_i^{(k)} - \vartheta_i^{(j)}$ for $i = 1, \ldots, n$, where $\vartheta^{(k)}$ and $\vartheta^{(j)}$ denote the joint configurations at two consecutive points on the path. $E(\Theta)$ is clearly coercive. Furthermore, since any rotation of a single angle $|\Delta_i^{(j)}|$ of more than π radians results in the same new angle as a change of $\Delta_i^{(j)} - 2\pi$, any minimum $\Theta^\star$ has to be such that $\vartheta_i^{(j)} - \vartheta_i^{(k)} \in [-\pi, \pi]$. Thus the total rotation of each angle after visiting all s points is at most πs at the minima. Further assuming there are a finite number of joints and points to be visited, this restricts $\Theta^\star$ to a compact space. Since $E(\Theta)$ is continuous on this space the optimization problem (3) has a solution.

The global minima are not unique if no restrictions are imposed in addition to the constraints in (3). Observe for example that if all the angles are changed by a multiple of 2π the function value will remain unchanged, as will the feasibility of the path. The objective function is not periodic along this line, however, since it remains constant with a change of any (equal and arbitrarilly small) size in all angels at once. In fact, the function is not periodic along any path since it is convex, as shown later on. It can be concluded from the non-uniqueness of the global minima that $E(\Theta)$ is not *strictly* convex.

Even when imposing the restriction $|\Delta_i^{(j)}| \leq \pi$ for $i = 1, \dots, n$ and $j = 1, \dots, s$ and requiring that a rotation of π radians is always done in the same direction, global minima need not be unique. This can be shown by a simple example. Lets say we have a robot with two arms of equal length, say $l_1 = l_2 = 1$, and the points on the path are $p^{(1)} = (2, 0)$, $p^{(2)} = (0, 0)$ and $p^{(3)} = (-2, 0)$. The path can only be obtained in the following manner: From $p^{(1)}$ to $p^{(2)}$ and from $p^{(2)}$ to $p^{(3)}$ the outer joint is rotated π radians, and from both $p^{(1)}$ to $p^{(3)}$ and $p^{(3)}$ to $p^{(1)}$ the inner joint is rotated π radians. The question is how to rotate the inner joint from $p^{(1)}$ to $p^{(2)}$; the rotation from $p^{(2)}$ to $p^{(3)}$ follows from this. It is clear that $|\Delta_1^{(1)}| = \pi/2$ minimizes $E(\Theta)$, giving the two possibilities $\Delta_1^{(1)} = \pm\pi/2$ for obtaining a global minimum. Thus global minima are in general not unique. Since these global minima are isolated from each other we have also shown that the problem is *not convex* under the aforementioned restrictions and specific constraints.

Generally there can be infinitely many paths through all the points, and finding the optimal one analytically is usually not possible. This encourages the use of a numerical optimization method. Much of the rest of this section and part of the next will analyze one such method applicable to (3) in detail.

To simplify notation and avoid being repetitive, $\vartheta^{(0)}$ and $\vartheta^{(s+1)}$ are used and are to be read as $\vartheta^{(s)}$ and $\vartheta^{(1)}$, respectively. $E(\Theta)$ can be rewritten as follows:

$$\begin{aligned} E(\Theta) &= E(\vartheta_1^{(1)}, \dots, \vartheta_n^{(1)}, \dots, \vartheta_1^{(s)}, \dots, \vartheta_n^{(s)}) \\ &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{s-1}(\vartheta_i^{(j+1)} - \vartheta_i^{(j)})^2 + \frac{1}{2}\sum_{i=1}^{n}(\vartheta_i^{(1)} - \vartheta_i^{(s)})^2 . \end{aligned} \tag{4}$$

This rewriting helps calculating the gradient, given by

$$\frac{\partial E}{\partial \vartheta_i^{(j)}} = 2\vartheta_i^{(j)} - (\vartheta_i^{(j+1)} + \vartheta_i^{(j-1)}) \qquad \text{for } i = 1, \dots, n \text{ and } j = 1, \dots, s . \tag{5}$$

With I being the $n \times n$ identity matrix, this results in the $n \times s$ Hessian matrix

$$\nabla^2 E = \begin{pmatrix} 2I & -I & & & & -I \\ -I & \ddots & \ddots & & & \\ & \ddots & & & & \\ & & & & \ddots & \\ & & & \ddots & \ddots & -I \\ -I & & & & -I & 2I \end{pmatrix} . \tag{6}$$

The Hessian is diagonally dominant and hermitian, implying that it is positive definite. Thus $E(\Theta)$ is convex, as mentioned before. The minimum is obviously zero and is obtained when all joint angles remain constant throughout the path, i.e. the arm does not move.

Now lets look at the constraints

$$c_j(\Theta) = F(\vartheta^{(j)}) - p^{(j)} = \begin{pmatrix} \sum_{k=1}^{n} l_k \cos \sum_{i=1}^{k} \vartheta_i^{(j)} - p_x^{(j)} \\ \sum_{k=1}^{n} l_k \sin \sum_{i=1}^{k} \vartheta_i^{(j)} - p_y^{(j)} \end{pmatrix} = 0 \quad \text{for } j = 1, \dots, s . \tag{7}$$

where $p_x^{(j)}$ and $p_y^{(j)}$ denote the x and y-coordinates of a point $p^{(j)}$. Separating each vector constraint into its x-component $c_{j,x}(\Theta)$ and y-component $c_{j,y}(\Theta)$, we get $2s$ scalar constraints in total.

The constraint gradients $\nabla c_{j,\gamma}(\Theta)$, where $\gamma \in \{x, y\}$, are vectors of length ns with zeros everywhere except possibly at indices $[(j-1)n+1, \ldots, jn]$. Denoting this part of the vector $\nabla c'_{j,\gamma}(\vartheta^{(j)})$, and using the notation

$$a_q^{(j)} = \sum_{k=q}^{n} l_k \cos \sum_{i=1}^{k} \vartheta_i^{(j)} \qquad \text{and} \qquad b_q^{(j)} = -\sum_{k=q}^{n} l_k \sin \sum_{i=1}^{k} \vartheta_i^{(j)}, \tag{8}$$

gives

$$\nabla c'_{j,x}(\vartheta^{(j)}) = \left(b_1^{(j)}, \ldots, b_n^{(j)}\right)^\mathsf{T}, \qquad \nabla c'_{j,y}(\vartheta^{(j)}) = \left(a_1^{(j)}, \ldots, a_n^{(j)}\right)^\mathsf{T}. \tag{9}$$

This gives the Jacobian matrix of constraints

$$\begin{aligned} A(\Theta)^\mathsf{T} &= \begin{pmatrix} \nabla c_{1,x}(\Theta) & \nabla c_{1,y}(\Theta) & \cdots & \nabla c_{s,x}(\Theta) & \nabla c_{s,y}(\Theta) \end{pmatrix} \\ &= \begin{pmatrix} \nabla c'_1(\vartheta^{(j)}) & & \\ & \ddots & \\ & & \nabla c'_s(\vartheta^{(j)}) \end{pmatrix}, \end{aligned} \tag{10}$$

where $\nabla c'_j(\vartheta^{(j)}) = \begin{pmatrix} \nabla c'_{j,x}(\vartheta^{(j)}) & \nabla c'_{j,y}(\vartheta^{(j)}) \end{pmatrix}$.

Due to the structure of $A(\Theta)$, where the blocks $\nabla c'_j(\vartheta^{(j)})$ have different non-zero segments, pairwise linear independence between $\nabla c(\Theta)_{j,x}$ and $\nabla c(\Theta)_{j,y}$ for $j = 1, \ldots, s$ implies linear independence between all the constraint gradients. If any of the constraint gradients are zero, they are not linearly indepedent. This is the case if, for some j, $\vartheta_1^{(j)} = k\pi/2$ for any $k \in \mathbb{Z}$ and $\vartheta_i^{(j)}$ for $i = 2, \ldots, n$ are multiples of π, i.e. when the robot arm is situated entirely on either the x-axis or the y-axis. They are also not linearly independent if, for some j, $\cos \vartheta_1^{(j)} = \cdots = \cos \sum_{i=1}^{n} \vartheta_i^{(j)}$ and $\sin \vartheta_1^{(j)} = \cdots = \sin \sum_{i=1}^{n} \vartheta_i^{(j)}$. This is the case if $\vartheta_i^{(j)}$ are multiples of π for $i = 2, \ldots, n$, i.e. when the robot arm is situated on *any* straight line. Otherwise the constraint gradients are linearly independent and thus LICQ holds, ensuring existence and uniqueness of Lagrange multipliers. The proof of linear independence is omitted for the sake of brevity.

The Hessian matrices of the constraint gradients are

$$\nabla^2 c'_{j,x} = -\begin{pmatrix} a_1 & a_2 & \cdots & a_n \\ a_2 & a_2 & & \\ \vdots & & \ddots & \\ a_n & & & a_n \end{pmatrix}, \qquad \nabla^2 c'_{j,y} = \begin{pmatrix} b_1 & b_2 & \cdots & b_n \\ b_2 & b_2 & & \\ \vdots & & \ddots & \\ b_n & & & b_n \end{pmatrix}. \tag{11}$$

Since all a_j and b_j can take on any value the constraint Hessian matrices are generally not positive semi-definite, and thus the constraints are, generally, not convex.

Next we look at adding the $2ns$ inequality constraints $-c \leq \vartheta_i^{(j)} \leq c$ for all the angles, with $0 < c < \pi$, to (3). Equivalently these constraints can be noted

$$d_i^{(j)} = c \pm \vartheta_i^{(j)} \geq 0 \qquad \text{for } i = 1, \ldots n \text{ and } j = 1, \ldots, s. \tag{12}$$

The gradients of the inequality constraints $\nabla d_i^{(j)}$ are vectors of length ns with zeros everywhere except at index $nj+i$. This value of this component is ± 1. For any fixed i and j the two constraints $d_i^{(j)}$ cannot be active simultaneously. However, if for some fixed j one of $d_i^{(j)}$ are active for all $i = 1, \ldots, n$, except possibly one, the constraint gradients ∇c_j and $\nabla d_i^{(j)}$ are not linearly independent. In fact, only (one of) the $d_i^{(j)}$ where either $b_i^{(j)}$ or $a_i^{(j)}$ are not zero need to be active (except possibly one of them) to prevent linear independence. And if the angles in any subset of $\vartheta_i^{(j)}$ are different only by multiples of π, the statement from the previous sentence holds when only considering the angles from the complementary subset. Since we have now described all the scenarios where the unit and negative unit vectors $\nabla d_i^{(j)}$ from the active inequality constraints can be combined linearly with each other to equal $\nabla c_{j,x}$ or $\nabla c_{j,y}$, or can be combined linearly with $\nabla c_{j,x}$ to equal $\nabla c_{j,y}$, LICQ holds if none of the scenarios are true.

Optimization method

The quadratic penalty method and the augmented Lagrangian were considered for solving the equality-constrained problem (3). The fact that LICQ holds for almost all configurations guarantees the existence of an optimal Lagrangian multiplier λ^* at a local minima, which is utilized by the augmented Lagrangian method. The augmented Lagrangian method also reduces the risk of the problem becoming ill conditioned since the penalty parameter isn't required to grow indefinitely. This is not the case for the quadratic penalty method. The advantage of the quadratic penalty method comes down to it being a simpler method to understand and implement, but this does not outweigh the benefits of the augmented Lagrangian method. The augmented Lagrangian method is therefore chosen.

The method is implemented as outlined in *Numerical Optimization* by Nocedal [1, p. 515]. The method relies on a separate method for approximating the minimizer of the augmented Lagrangian function, $\mathscr{L}_A(\Theta_k, \lambda^k, \mu_k)$, as defined by

$$\mathscr{L}_A(\Theta, \lambda, \mu) = E(\Theta) - \sum_{i\in\epsilon} \lambda_i c_i(\Theta) + \frac{\mu}{2} \sum_{i\in\epsilon} c_i^2(\Theta) \tag{13}$$

This function changes in every iteration as the parameters are updated by the method. An appropriate method for conducting unconstrained optimization of $\mathscr{L}_A(\Theta_k, \lambda^k, \mu_k)$ at each iteration is therefore required. The BFGS method is chosen for this purpose. The BFGS method does not require an expression for the Hessian of the objective function. One therefore avoids the challenges of calculating and implementing the Hessian of the augmented Lagrangian function. The calculation and implementation of the gradient of the augmented Lagrangian function is required, but can be done with relative ease using the theory from the previous section.

Updating the Lagrangian multipliers is done by utilizing the satisfied KKT conditions to make a good guess at the optimal λ^*. This can be expressed by the update equation $\lambda_i^{k+1} = \lambda_i^{k+1} - \mu_k c_i(\Theta)$. The optimal sequences μ_k and τ_k depend on the specific problem, and the parameters can simply be tweaked until satisfactory performance is observed. An adaptive approach to the determination of these sequences could have been taken, but implementation of this was considered to not be worth the time and effort required.

The value of $\|\nabla_\Theta \mathscr{L}_A(\Theta,\lambda,\mu)\|$ being less than a given tolerance τ_G is chosen as the final convergence criterion. This turned out to work well in practice. One could also consider the norm of the difference between the minimizers of $\mathscr{L}_A(\Theta_k,\lambda^k,\mu_k)$ and $\mathscr{L}_A(\Theta_{k+1},\lambda^{k+1},\mu_{k+1})$, but this was not implemented since the other criterion proved to be sufficient for the problems considered. The method behaves well with the zero-vector initial Lagrange multiplier. Generating better initial guesses was therefore not prioritized. An attempt was made to generate good guesses for the initial configuration of the joints. This was done using unconstrained optimization to make sure the method started off in a feasible point. This turned out to be counter-productive due to this increasing the method's tendency to converge towards a solution that could easily be shown not to be optimal.

When introducing the angular constraints defined in (12) it becomes challenging to describe the configuration space $\mathscr{C}$ in general. One approach is to systematically generate a large amount of points from feasible configurations. The number of points can easily be increased or decreased to change the resolution of $\mathscr{C}$. Let us call this set of points $\mathscr{P}$. Then one possibility is to check whether a circular neighborhood of some radius r, centered at a point $p^{(j)}$ the robot arm should visit, contains a specific number n of points from $\mathscr{P}$. The challenge is to find a balance for r and n such that as many points as possible in $\mathscr{C}$ are accepted, without accepting too many points from outside $\mathscr{C}$.

Our implementation focuses mainly on accepting points in $\mathscr{C}$, so that there is a considerable possibility of accepting points that are not in $\mathscr{C}$. However, it is usually clear from observation of a plot of the configurations space and the point in question if the point belongs to $\mathscr{C}$ or not, at least if the point is not relatively close to the border of the domain. Thus one can turn to plotting if there is reason to doubt the results of the algorithm checking the neighborhood of the points. In the next section an example that makes this clearer is considered. If a point is not considered to be in $\mathscr{C}$ by this method, the program is terminated before any optimization method is initialized.

Solving the optimization problem (3) subject to both inequality constraints and equality constraints complicates the problem and requires a new approach. The inequality constraints are of the form $l \le \vartheta_i^{(j)} \le u$, known as bound constraints, and methods designed for constraints of this kind, like the bound-constrained augmented Lagrangian can be considered. Sequential quadratic programming methods should also be considered. The barrier approach, introduced in *Numerical Optimization* [1, ch. 19.1, 19.6], also emerges as an alternative. Taking the latter approach and combining it with the augmented Lagrangian method enables us to take advantage of the knowledge and experience obtained when solving the equality-constrained problem. There's also the added benefit of being able to re-use code and ideas from the equality-constrained problem .

The slack variable s of length $2ns$ and the penalty parameter σ is introduced and a new variable Φ is defined as $\left(\Theta,\ s\right)^\intercal$. The barrier approach involves defining the objective function as $B(\Phi,\sigma) = E(\Theta) - \sigma\sum_i^m \log s_i$ and expressing all constraints as equality constraints. The number of inequality constraints in the original problem is denoted by $m = 2ns$. The equality constraints remains the same as before and the inequality constraints are changed to equality constraints defined as $d_i^{(j)} - s_i = 0$.

The problem can now be approached as an equality-constrained problem, and the augmented

Lagrangian method can be used. The augmented Lagrangian function becomes

$$\mathscr{L}_A(\Phi,\sigma,\lambda,\mu) = B(\Phi,\sigma) - \sum_i^q \lambda c_i(\Phi) + \frac{\mu}{2}\sum_i^q c_i(\Phi)^2 \tag{14}$$

where $q = 2s(1+n)$ is the total number of constraints. The problem is solved in essentially the same way as before. The BFGS method is still used for solving the subproblem, and gradients of objectives and constraints are calculated and implemented. The new penalty parameter σ must be tweaked in order to obtain optimal performance. In this case the convergence test mentioned earlier, with the norm of the difference between two subsequent configurations, is applied. This is combined with a convergence test identical to the one used earlier. The former test was used due to the poor practical performance of the latter one.

Numerical experiments

In this section the numerical properties of the methods described above are investigated. First equality constraints as in (3) are considered. The problem parameters chosen for the following numerical analysis are

$$\begin{aligned} &l_1 = 3,\ l_2 = 2,\ l_3 = 2, \\ &p^{(1)} = (5,0),\ p^{(2)} = (4,2),\ p^{(3)} = (6,0.5),\ p^{(4)} = (4,-2),\ p^{(5)} = (5,-1). \end{aligned} \tag{15}$$

The method calculates the configurations shown in figure 1. One can observe from the plots that the equality-constraints are satisfied. This sequence of configurations gives the result $E(\Theta) = 0.905$. With other initial guesses Θ_0, resulting in other configurations, $E(\Theta)$ has taken on higher values. This indicates that stationary points that are not global minima exist. The initial guess resulting in the lowest value of $E(\Theta)$ with these parameters was simply the zero-vector. The method has also been run with other problem parameters, and has converged towards feasible solutions in every case tested.

Since there are many feasible paths, and the global solution is not known, it is not possible to test convergence to an analytical or approximately correct numerical reference solution. It is, however, possible to analyze how fast the method converges towards the final solution of a specific problem instance. The result of such an analysis, with parameters (15), is presented in figure 2. In addition the value of the objective function and a measure of total distance to the destination points at each iteration are plotted. The value of the objective function is increasing in this case. This can be interpreted as the cost of coming closer to satisfying the equality constraints.

The problem

$$\begin{aligned} &l_1 = 3,\ l_2 = 2,\ l_3 = 2, \\ &p^{(1)} = (-1,5),\ p^{(2)} = (-3,3),\ p^{(3)} = (-3,-4),\ p^{(4)} = (0,5),\ p^{(5)} = (3,2). \end{aligned} \tag{16}$$

is considered as the main example in the numerical analysis of the method for solving problems with inequality constraints of type (12). First the configuration space is analyzed using the method described in the last section. The result is visualized in figure 3.

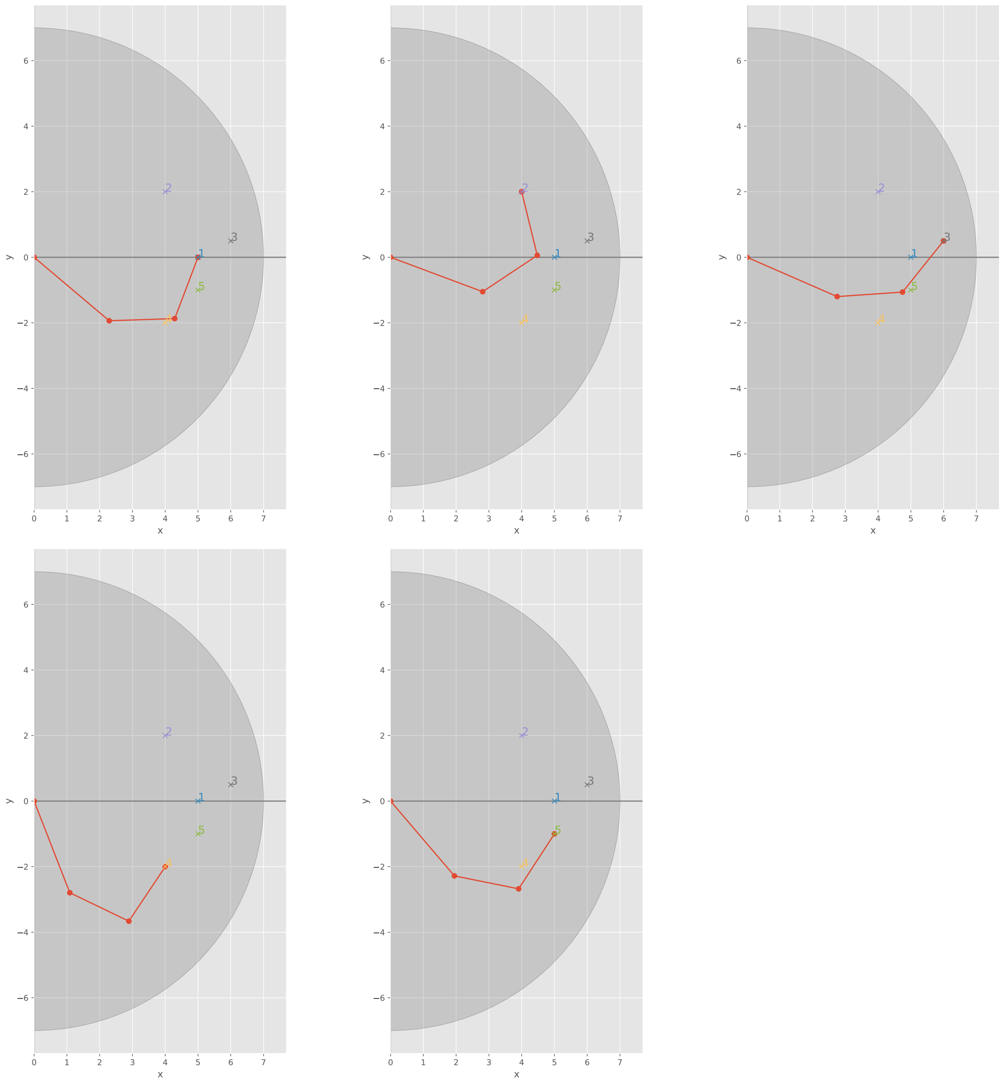

Figure 1: The configurations of the robot arm calculated by the augmented Lagrangian method in combination with BFGS. The plots give visual confirmation the end-effector coinciding with the destination points.

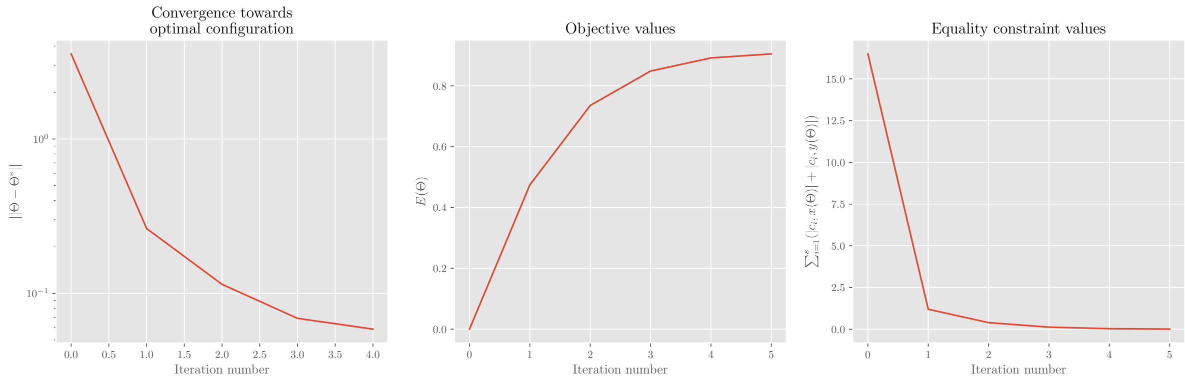

Figure 2: Analysis of the optimization method solving problem (15). The leftmost plot shows how the method convergences towards the final numerical solution for the problem. The middle plot shows the values of the objective function for each method iteration. The rightmost plot shows the values of the equality constraints for each iteration.

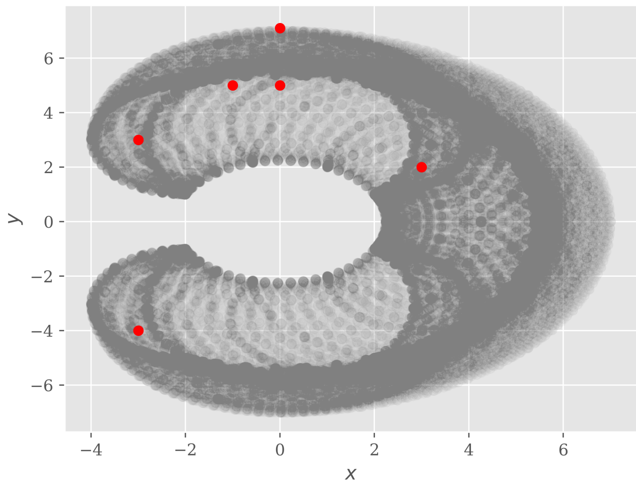

Figure 3: Numerical sketch of the configuration space $\mathscr{C}$ of the robot arm with parameters as stated in (16) and inequality constraints (12), with $c = \pi/2$. The red points are the points on the path, in addition to the point $(0, 7.1) \notin \mathscr{C}$.

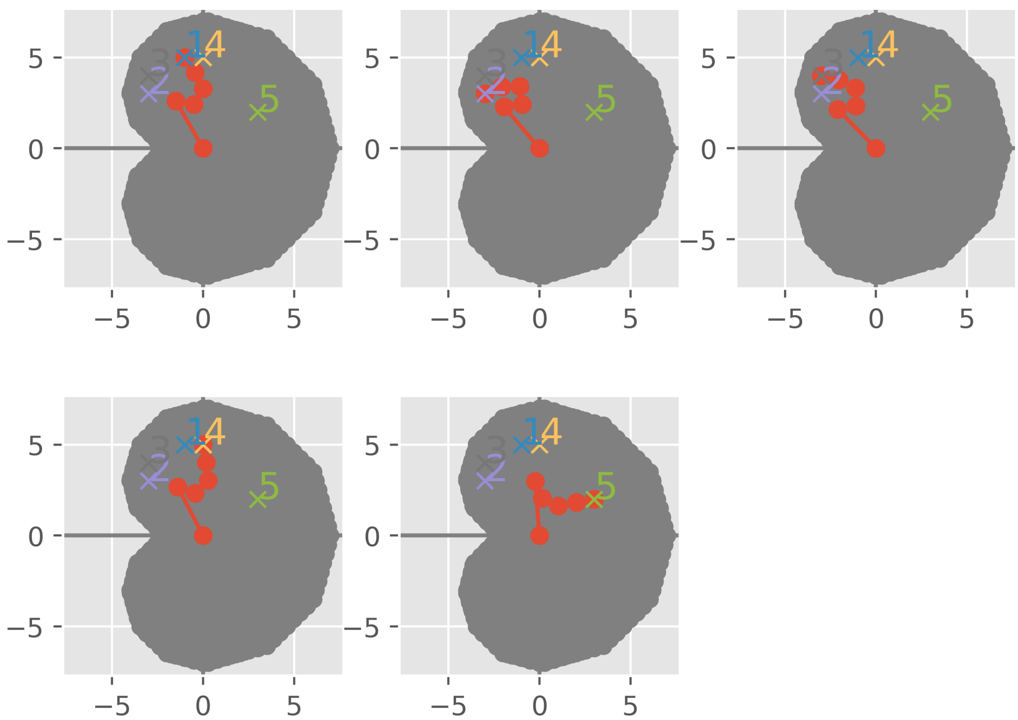

Figure 4: The configurations of the robot arm when attempting to satisfy both equality- and inequality constraints. Only the equality constraints is satisfied in the end. The solution is calculated using a log-barrier method in combination with augmented Lagrange method and BFGS.

The points $p^{(j)}$, for $j = 1, \dots, 5$, red in the figure, are all accepted as points belonging to the configuration space $\mathscr{C}$, as desired. The sixth point, $p^{(6)} = (0, 7.1)$, is obviously not reachable since the length of the arm is 7. Despite this fact it is accepted by the algorithm. However, as mentioned in the theory, it is easy to conclude that the point is not reachable by inspection of the plot in figure 3. If the point is changed to $(0, 7.2)$, it is not accepted, and the optimization algorithm terminates.

Minimizing the problem subject to inequality constraints as well as equality constraints proved to be challenging. The method produces a solution to the problem in some cases, but fails to satisfy the inequality constraints in almost every case. The intuitive solution to this problem is to increase the penalty parameter σ, but this very quickly caused the method to get stuck in a section of the BFGS method. The method also struggles with situations where elements of the slack variable s were set to almost-zero or negative values. This produced problems during the evaluation of the augmented Lagrangian and it's gradient. All attempts at fixing this error resulted in the method getting stuck during minimization of a subproblem. A plot showing a solution and it's associated configuration space is displayed in figure 4.

Conclusion

In the case for the equality-constrained problem the augmented Lagrangian method, utilizing the quasi-Newton method BFGS to solve the subproblems, seems to generate nice results for every set of problem parameters tested. It also generally converges in few iterations. The BFGS method used to find minimizers of the subproblems also generally converges quickly.

The algorithm for creating points to approximate the configuration space $\mathscr{C}$ of the robot arm when constraints restricting the freedom of the rotation of each joint are introduced, also runs as desired. It is straightforward to decide whether a destination point is inside this space or not visually from plots of $\mathscr{C}$ and the point, as long as the point is not relatively close to the boundary of the space.

The implementation of the log-barrier method is unsuccessful. The augmented Lagrangian method with underlying log-barrier objective rarely produces solutions satisfying the convergence criterion. When reducing the parameter controlling the penalty for breaking the inequality constraint enough, the method behaves better. To get results of any interest without encountering errors, the parameter needs to be reduced so much that the inequality constraints are essentially not taken into account at all. This leaves us with a method that almost never satisfies the inequality constraints.

The code related to the algorithms and numerical experiments can be found at `https://github.com/JakobGM/robotarm-optimization`.

To summarize, the attempt at solving (3) with equality constraints was successful. When considering the problem subject to the inequality constraints defined in (12), the method failed. This is probably due to some kind of error in the implementation. Alternatively some function or gradient is calculated incorrectly, or the method or problem is not understood fully by the group. More time would have been required in order to remedy this.

References

[1] J. Nocedal, *Numerical Optimization*, Springer, 2006.

[2] D. of Mathematical Sciences, *Student Project 2, TMA4180*.