Model-Free Safe Exploration and Optimization

Master's Thesis

Yang Wang

Department of Electrical and Computer Engineering
Chair of Control Systems

Supervisors:
Prof. Dr.-Ing. Daniel Görges
Prof. Dr.-Ing. Giancarlo Ferrari Trecate
Doctoral Assistant Baiwei Guo

March 31, 2022

Abstract

For many optimization problems in applications ranging from science, engineering, to economics, etc., objectives and constraints may not be explicitly known by mathematical expressions, but can only be accessed as outputs of black-boxes, such as simulations or experiments. The fundamental ingredient in traditional optimization methods, the derivative information, is thus not available. Such a setting encourages the use of model-free optimization methods, which do not require the availability of system models. Instead, these methods determine the next point to evaluate using previous input and output information from the systems. The choice of the next evaluation point is the essence of model-free optimization methods. In addition to the model unavailability issue, in many real-world applications, safety and the ability to operate under various constraints are critical prerequisites. The main concern of safety lies in the issue that the operation of systems in constraint-violated conditions may either not be able to perform or cause severe consequences such as device damages or human injuries. The safety requirement is typically crucial in applications such as medical treatments, power system operations, or robotic platforms. In model-free optimization scenarios, the safety constraints require the optimizer never evaluate decision points outside of certain safety areas. Recently, some safety-aware model-free optimization methods are proposed, such as the safe Bayesian optimization (SafeOpt-MC) [BKS21] and the safe logarithmic barrier method (0-LBM) [UKK19]. Though the SafeOpt-MC method enjoys evaluation-efficiency, it suffers from a high computational cost and limited scalability to high dimensional problems. The 0-LBM method on the other hand hardly approaches an optimal point when it is close to a boundary due to exponentially decreasing step sizes. In this thesis, we propose a novel model-free safe optimization algorithm, the safe-line-search optimization (0-SLS). This method extends the conventional line-search method to account for model-free, safety-critical circumstances. Given an initial safety-feasible point, the algorithm minimizes the objective function and evaluates only points that satisfy all safety constraints. To this end, with a low computational cost the algorithm carefully searches along a descent direction utilizing smoothness properties of underlying objective and constraint functions. The algorithm is empirically tested on several numerical examples and presents general outperformance than existing safe optimization methods. Finally, the algorithm is applied to a large-scale AC Optimal Power Flow optimization problem, and it demonstrates a comparable performance to the benchmark results. The code of the algorithm is available[1].

Keywords: Model-free optimization, Derivative-free optimization, Safe optimization, Line-search method, Power system, Optimal power flow problem.

[1] An example code is available at: https://github.com/yanggwang/safe_line_search_optimization

Sample from Chapter 2

2 Safe-line-search optimization

2.1 Problem formulation

The model-free safe optimization problem is the problem in which both objective and constraint functions are unknown, and the constraints are safety-critical, which implies that no evaluation violating safety constraints throughout the optimization process shall be made. The model-free safe optimization can be formulated as:

Problem 2. ***Model-Free Safe Optimization Problem***

$$\underset{\boldsymbol{x}\in\mathbb{R}^d}{\text{minimize}} \quad f^0(\boldsymbol{x}) \tag{2.1}$$

$$\text{subject to} \quad f^i(\boldsymbol{x}) \leq 0,\ i = 1, \ldots, m \tag{2.2}$$

where $\boldsymbol{x} \in \mathbb{R}^d$ is the decision variable, the objective function $f^0(\boldsymbol{x}) : \mathbb{R}^d \to \mathbb{R}$ and the constraints $f^i(\boldsymbol{x}) : \mathbb{R}^d \to \mathbb{R}$ are not explicitly known and are possibly non-linear, non-convex functions. d is the number of dimensions of decision variable and m is the number of constraints.

In problem 2, both objective and constraint functions can only be evaluated in a safety feasible set $\mathcal{D}$. The safety feasible set $\mathcal{D}$ is defined as:

$$\mathcal{D} := \{\boldsymbol{x} \in \mathbb{R}^d : f^i(\boldsymbol{x}) \leq 0, i = 1, \ldots, m\} \tag{2.3}$$

2.2 Assumptions

For solving model-free optimization problem, some assumptions about the objective and constraint functions are assumed.

Assumption 1. *Let $L, M \geq 0$, the constraints $f^i(\boldsymbol{x})$ for $i = 1, \ldots, m$ are L-Lipschitz continuous on $\mathcal{D}$ and the objective and constraint functions $f^i(\boldsymbol{x})$ for $i = 0, \ldots, m$ are M-smooth on $\mathcal{D}$.*

Assumption 2. *The feasible set $\mathcal{D}$ is not empty, and a known safety feasible starting point $\boldsymbol{x}_0$ is given for which $f^i(\boldsymbol{x}_0) \leq 0$, for $i = 1, \ldots, m$.*

2.3 Preliminaries

In this thesis, the following notations and definitions are used:

- Denote by $\|\cdot\|_1$ the l_1-norm on $\mathbb{R}^d$;
- Denote by $\|\cdot\|_2$ the l_2-norm on $\mathbb{R}^d$;

Definition 1. *A function $f(\boldsymbol{x})$ is called L-Lipschitz continuous on $\mathbb{R}^d$ with $L \geq 0$ if for all $\boldsymbol{x}$, $\boldsymbol{y} \in \mathbb{R}^d$:*

$$|f(\boldsymbol{y}) - f(\boldsymbol{x})| \leq L\|\boldsymbol{y} - \boldsymbol{x}\|_2 \tag{2.4}$$

Definition 2. *A function $f(\boldsymbol{x})$ is called M-smooth on $\mathbb{R}^d$ with $M \geq 0$ if the gradient $\nabla f(\boldsymbol{x})$ is M-Lipschitz continuous on $\mathbb{R}^d$:*

$$\|\nabla f(\boldsymbol{y}) - \nabla f(\boldsymbol{x})\|_2 \leq M\|\boldsymbol{y} - \boldsymbol{x}\|_2 \tag{2.5}$$

Definition 3 (sub-Gaussian random variable)**.** *A random variable X is called sub-Gaussian (subG) with variance σ^2, if $\mathbb{E}(X) = 0$ and for any $s \in \mathbb{R}$ its moment generating function satisfies:*

$$\mathbb{E}\left[\exp(sX)\right] \leq \exp(\frac{\sigma^2 s^2}{2}) \tag{2.6}$$

In optimization problems, the mathematical tool used to analyze the minimizers of smooth functions and bounds conditions for constraints is Taylor's theorem. It is the central tool for studying the safety constraints in this thesis. Taylor's theorem states the following.

Theorem 1 (**Taylor's theorem**)**.** *Let $f^i(\boldsymbol{x})$ be a continuously differentiable, and $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, there is:*

$$f^i(\boldsymbol{y}) = f^i(\boldsymbol{x}) + \nabla f^i\left(\boldsymbol{x} + t\left(\boldsymbol{y} - \boldsymbol{x}\right)\right)^{\mathrm{T}}\left(\boldsymbol{y} - \boldsymbol{x}\right) \tag{2.7}$$

where $t \in (0, 1)$. If $f^i(\boldsymbol{x})$ is twice continuously differentiable, then:

$$f^i(\boldsymbol{y}) = f^i(\boldsymbol{x}) + \nabla f^i\left(\boldsymbol{x}\right)^{\mathrm{T}}\left(\boldsymbol{y} - \boldsymbol{x}\right) + \frac{1}{2}(\boldsymbol{y} - \boldsymbol{x})^{\mathrm{T}}\nabla^2 f^i\left(\boldsymbol{x} + t\left(\boldsymbol{y} - \boldsymbol{x}\right)\right)\left(\boldsymbol{y} - \boldsymbol{x}\right) \tag{2.8}$$

for $t \in (0, 1)$.

From Theorem 1, if $t = 0$ and using Assumption 1 about the function $f^i(\boldsymbol{x})$ to replace $\nabla f^i(\boldsymbol{x})$ and $\nabla^2 f^i(\boldsymbol{x})$ by L and M respectively, then from (2.7) there is:

$$f^i(\boldsymbol{y}) \leq f^i(\boldsymbol{x}) + L\|\boldsymbol{y} - \boldsymbol{x}\|_2 \tag{2.9}$$

and from (2.8), there is:

$$f^i(\boldsymbol{y}) \leq f^i(\boldsymbol{x}) + \nabla f^i(\boldsymbol{x})^{\mathrm{T}}(\boldsymbol{y} - \boldsymbol{x}) + \frac{1}{2}M\|\boldsymbol{y} - \boldsymbol{x}\|_2^2 \tag{2.10}$$

Equations (2.9) and (2.10) will be used frequently in this thesis.

To determine if or not a point $\boldsymbol{x}_*$ is an (local) optimum point for the optimization problem, an optimality condition is used.

In constrained optimization problems, the necessary condition for $\boldsymbol{x}_*$ to be a (local) optimum of the problem is the *Karush-Kuhn-Tucker (KKT) condition*. This condition is formulated based on the *Lagrangian function* which is defined below:

Definition 4. *The Lagrangian function $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ for the constrained optimization problem 2 is defined as:*

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f^0(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda^i f^i(\boldsymbol{x}) \tag{2.11}$$

where $\boldsymbol{\lambda} \in \mathbb{R}_+^m$ is called the Lagrangian multipliers.

Using the Lagrangian function (2.11), the constrained optimization problem 2 is converted into an unconstrained problem:

$$\underset{\boldsymbol{x},\ \boldsymbol{\lambda}}{\text{minimize}}\ \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) \tag{2.12}$$

The optimality condition for constrained optimization problem 2 and the equivalent unconstrained optimization problem (2.12) is formulated as the KKT condition.

The KKT condition holds in the presence of a regularity condition of *Linear independence constraint qualification (LICQ)*. It is defined as:

Definition 5. *The linear independence constraint qualification (LICQ) for the safe optimization problem 2 holds if constraint gradients $\nabla f^i(\boldsymbol{x})$, $i = 1, \ldots, m$ are linearly independent.*

With the definitions 4 and 5, the KKT condition is defined as:

Theorem 2 (**Karush-Kuhn-Tucker condition**). *Suppose that* $f^i(\boldsymbol{x})$, $i = 0, \ldots, m$ *are continuously differentiable, and that the LICQ holds for constraints* $f^i(\boldsymbol{x})$, $i = 1, \ldots, m$. *A point* $\boldsymbol{x}_*$ *is a local minimizer of the objective function* $f^0(\boldsymbol{x})$ *if there exists a set of Lagrangian multipliers* λ_*^i, $i = 1, \ldots, m$ *such that the following conditions hold:*

$$\|\nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}_*, \lambda_*^i)\|_2 = 0, \ \forall i = 1, \ldots, m \tag{2.13}$$

$$f^i(\boldsymbol{x}_*) \leq 0, \ \forall i = 1, \ldots, m \tag{2.14}$$

$$\lambda_*^i f^i(\boldsymbol{x}_*) = 0, \ \forall i = 1, \ldots, m \tag{2.15}$$

$$\lambda_*^i \geq 0, \ \forall i = 1, \ldots, m \tag{2.16}$$

In Theorem 2, (2.13) is the stationary equation, (2.14) defines the primary feasibility, (2.15) is the complementary slackness and (2.16) defines the dual feasibility.

The KKT condition is a first-order necessary optimality condition as it considers only the properties of the gradients of the objective and constraint functions. A point that satisfies the KKT condition is called a KKT point.

With the required preliminaries, a model-free safe optimization algorithm called the Safe-Line-Search (SLS) method is proposed. This algorithm considers two situations: first, the function measurements are exact (e-SLS); second, the function measurements are corrupted by noises (n-SLS). In the following sections, algorithms for each case are formulated.

2.4 Main idea of the algorithm

The model-free safe-line-search algorithm employs a line search optimization scheme using estimated gradients of unknown objective and constraint functions while satisfying safety constraints. In this algorithm, at each iteration, gradients of unknown functions are estimated using finite-difference method with a safety-guarantee difference step length, and a descent search direction is computed using the gradient estimators. Along the search direction, a step length is carefully chosen such that a sufficient decrease in the objective can be expected and all safety constraints will not be violated. The iteration is repeated until a convergence condition is satisfied. The safe-line-search algorithms can be summarized by the following steps:

1. Estimate function gradients;
2. Find a search direction;
3. Compute a safety-guarantee step length;
4. Update new decision point and check the convergence condition;

The algorithm is first formulated for the case of exact measurements, then is extended to the case of noisy measurements with modifications to deal with uncertainties involved in function measurements. Section 2.5 introduces the safe-line-search algorithm for exact measurements. Section 2.6 describes the modifications of algorithm for the case of noisy measurements. Section 2.7 discusses the convergence and safety properties of the algorithm.

2.5 Safe-line-search algorithm for exact measurements (e-SLS)

2.5.1 Section structure

The structure of this section follows the formulation step of the algorithm. The safe-line-search algorithm firstly estimates the gradients of the underlying functions, then a search direction is computed according to estimated gradients. A safe set is then defined so that the next point within this set is safety guarantee. If the optimization point is close to a boundary, the initial search direction is projected onto the perpendicular direction of gradients of the active constraints. Finally, a step length is selected along the computed search direction within the formulated safe set. The iteration is repeated until a convergence condition is met.

2.5.2 Gradient estimator

In model-free optimization problem, the objective and constraint functions are only accessible as function values returned from a black-box simulation or experiment. The important information for optimization, the function gradient, is not directly available. While, the gradients of unknown functions can be estimated using functions measurements by *finite difference method*. Concretely, at point $\boldsymbol{x}_k \in \mathbb{R}^d$, the gradient of a multivariate function $\nabla f^i(\boldsymbol{x}_k)$ can be estimated by:

$$G^i(\boldsymbol{x}_k, \nu_k) = \sum_{j=1}^{d} \frac{f^i(\boldsymbol{x}_k + \nu_k \boldsymbol{e}_j) - f^i(\boldsymbol{x}_k)}{\nu_k} \boldsymbol{e}_j \tag{2.17}$$

where $G^i(\boldsymbol{x}_k, \nu_k)$ is the gradient estimator for function $f^i(\boldsymbol{x}_k)$, ν_k is the difference length, $\boldsymbol{e}_j$ is the j-th coordinator vector.

As the gradient estimator is not identical to the true gradient, the estimation deviation between the estimator and true gradient exists. Though in many applications, the estimated gradients are accurate enough to provide a good solution, in safety-critical optimization, however, the deviation of gradient estimator from the true one shall be considered with much cares, since an inaccurate gradient information may lead to safety violation in function evaluations.

Denote by Δ_k^i the difference between the gradient estimator $G^i(\boldsymbol{x}_k, \nu_k)$ and the true gradient $\nabla f^i(\boldsymbol{x}_k)$:

$$\Delta_k^i = G^i(\boldsymbol{x}_k, \nu_k) - \nabla f^i(\boldsymbol{x}_k) \tag{2.18}$$

With Assumption 1, the following holds:

Lemma 1 ([UKK19]). *The norm of estimation deviation* Δ_k^i *(2.18) is upper bounded by:*

$$\|\Delta_k^i\|_2 = \|G^i(\boldsymbol{x}_k, \nu_k) - \nabla f^i(\boldsymbol{x}_k)\|_2 \leq \frac{\sqrt{d}M\nu_k}{2} \tag{2.19}$$

Proof. *According to the assumption 1 of M-smooth on function* $f^i(\boldsymbol{x})$*, there is:*

$$f^i(\boldsymbol{x}_k + \nu_k \boldsymbol{e}_j) \leq f^i(\boldsymbol{x}_k) + \nabla f^i(\boldsymbol{x}_k)^{\mathrm{T}}(\nu_k \boldsymbol{e}_j) + \frac{1}{2}M\|\nu_k \boldsymbol{e}_j\|_2^2 \tag{2.20}$$

using (2.17) and (2.18), we have:

$$\|\Delta_k^i\|_2 = \sqrt{\sum_{j=1}^{d}\left[\frac{f^i(\boldsymbol{x}_k + \nu_k \boldsymbol{e}_j) - f^i(\boldsymbol{x}_k)}{\nu_k} - \nabla f^i(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{e}_j\right]^2 \|\boldsymbol{e}_j\|_2^2} \tag{2.21}$$

$$= \sqrt{\sum_{j=1}^{d}\left[\frac{f^i(\boldsymbol{x}_k + \nu_k \boldsymbol{e}_j) - f^i(\boldsymbol{x}_k) - \nabla f^i(\boldsymbol{x}_k)^{\mathrm{T}}(\nu_k \boldsymbol{e}_j)}{\nu_k}\right]^2} \tag{2.22}$$

$$\leq \sqrt{\sum_{j=1}^{d}\left[\frac{M\|\nu_k \boldsymbol{e}_j\|_2^2}{2\nu_k}\right]^2} \tag{2.23}$$

$$= \frac{\sqrt{d}M\nu_k}{2} \tag{2.24}$$

□

It can be seen that the estimation error $\|\Delta_k^i\|_2$ depends on the step length ν_k, the dimension of function d and the smoothness constant M. As for a certain application, the dimension d is fixed, and the smoothness property is determined by the underlying function. The estimation error can be handled by adapting the step length ν_k.

Given an acceptable upper bound μ on the gradient estimation deviation $\|\Delta_k^i\|_2$:

$$\|\Delta_k^i\|_2 \leq \mu \tag{2.25}$$

the step length satisfying the estimation accuracy can be computed by:

$$\nu_k = \frac{2\mu}{\sqrt{d}M} \tag{2.26}$$

In addition to the gradient estimation accuracy, the safety requirement shall be fulfilled while estimating the gradient. That is, the chosen step length ν_k is such that the function evaluations at $\boldsymbol{x}_k + \nu_k \boldsymbol{e}_j$ do not violate the safety constraints.

Given the *Lipschitz* assumption 1 about the constraint functions $f^i(\boldsymbol{x})$, a safety guaranteed step length can be computed by the following.

Lemma 2 ([UKK19]). *For* $\nu_k \leq \frac{\min\limits_{i=1,\ldots,m}\{-f^i(\boldsymbol{x}_k)\}}{2L}$, *the evaluations made around feasible point at* $\boldsymbol{x}_k + \nu_k \boldsymbol{e}_j$ *are safety feasible.*

Proof. *For any point* $\boldsymbol{y} \in \mathbb{R}^d$ *satisfying* $\|\boldsymbol{y} - \boldsymbol{x}_k\|_2 \leq \frac{\min\limits_{i=1,\ldots,m}\{-f^i(\boldsymbol{x}_k)\}}{2L}$, *there is:*

$$
\begin{aligned}
f^i(\boldsymbol{y}) &\leq f^i(\boldsymbol{x}_k) + \nabla f^i(\boldsymbol{x}_k)^{\mathrm{T}}(\boldsymbol{y} - \boldsymbol{x}_k) && (2.27)\\
&\leq f^i(\boldsymbol{x}_k) + L\|(\boldsymbol{y} - \boldsymbol{x}_k)\|_2 && (2.28)\\
&\leq f^i(\boldsymbol{x}_k) + L\frac{-f^i(\boldsymbol{x}_k)}{2L} && (2.29)\\
&\leq \frac{1}{2} f^i(\boldsymbol{x}_k) \leq 0 && (2.30)
\end{aligned}
$$

□

Considering both the estimation accuracy and safety constraints, the step length ν_k for computing the gradient estimator is chosen as:

$$
\nu_k = \min\left\{ \frac{2\mu}{\sqrt{d}M}, \frac{\min\limits_{i=1,\ldots,m}\{-f^i(\boldsymbol{x}_k)\}}{2L} \right\} \tag{2.31}
$$

With the computed gradient estimator $G^i(\boldsymbol{x}_k, \nu_k)$, the search direction for optimization can be determined.

2.5.3 Search direction

The search direction is the direction along which the objective function is decreased. It can be determined according to the gradient of the objective function. With the gradient information, there are two common search directions:

- The steepest direction;
- The *Newton direction*.

The steepest direction is the most obvious search direction, which is defined as the direction of negative gradient of the objective function. Among all directions one can move from $\boldsymbol{x}_k$, the steepest direction is the one along which the objective function decreases most rapidly. Let $\boldsymbol{p}_k \in \mathbb{R}^d$ be a search direction. The steepest direction can be written as:

$$
\boldsymbol{p}_k^s = -\nabla f^0(\boldsymbol{x}_k) \tag{2.32}
$$

where $\nabla f^0(\boldsymbol{x}_k)$ is the gradient of the objective function at point $\boldsymbol{x}_k$.

Though the steepest direction has an advantage that it requires only the first

order gradient of the objective function, it may be slow for difficult problems.

The *Newton direction* considers the second-order information of the objective function. It is derived from the second-order Taylor's theorem 1. If $\boldsymbol{p}_k$ is a search direction at point $\boldsymbol{x}_k$, then the function value at $\boldsymbol{x}_k + \boldsymbol{p}_k$ can be estimated by:

$$f^0(\boldsymbol{x}_k + \boldsymbol{p}_k) \approx f^0(\boldsymbol{x}_k) + \nabla f^0(\boldsymbol{x})^{\mathrm{T}} \boldsymbol{p}_k + \frac{1}{2} \boldsymbol{p}_k^{\mathrm{T}} \nabla^2 f^0(\boldsymbol{x}_k) \boldsymbol{p}_k \tag{2.33}$$

If the Hessian $\nabla^2 f^0(\boldsymbol{x}_k)$ at $\boldsymbol{x}_k$ is positive definite, the *Newton direction* is defined as the vector $\boldsymbol{p}_k$ that minimizes function $f^i(\boldsymbol{x}_k + \boldsymbol{p}_k)$. The solution of this problem can be computed by setting the derivative of $f^0(\boldsymbol{x}_k + \boldsymbol{p}_k)$ to zero. By this computation, the *Newton direction* $\boldsymbol{p}_k^N$ is defined as:

$$\boldsymbol{p}_k^N = - \left(\nabla^2 f^0(\boldsymbol{x}_k) \right)^{-1} \nabla f^0(\boldsymbol{x}_k) \tag{2.34}$$

The *Newton direction* has an advantage of fast local convergence. Once reaching to a neighborhood of the solution, the optimal point can be found in only a few iterations. The main drawback of *Newton direction* is the requirement of Hessian $\nabla^2 f^0(\boldsymbol{x}_k)$ and its positive definite. In model-free optimization problem, the estimation of second-order gradients can be computationally expensive and sensitive to measurement errors.

An attractive alternative to *Newton direction* is the *Quasi-Newton direction*. It maintains the superlinear convergence property of the *Newton direction* while does not require computation of Hessian. The *Quasi-Newton direction* replaces $\nabla^2 f^0(\boldsymbol{x}_k)$ by an approximated Hessian matrix $\boldsymbol{Y}_k$, and iteratively updates the approximation using information obtained from previous iterations.

The idea behind the Hessian approximation is that the changes in the gradient $\nabla f^0(\boldsymbol{x}_k)$ provides information about the second-order derivative of the objective function.

By the definition of second-order gradient, we know that:

$$\nabla^2 f^0(\boldsymbol{x}_k)(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) \approx \nabla f^0(\boldsymbol{x}_k) - \nabla f^0(\boldsymbol{x}_{k+1}) \tag{2.35}$$

The approximated Hessian matrix $\boldsymbol{Y}_{k+1}$ satisfies Equation (2.35) as:

$$\boldsymbol{Y}_{k+1} \boldsymbol{s}_k = \boldsymbol{g}_k \tag{2.36}$$

where

$$\boldsymbol{s}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \tag{2.37}$$

$$\boldsymbol{g}_k = \nabla f^0(\boldsymbol{x}_{k+1}) - \nabla f^0(\boldsymbol{x}_k) \tag{2.38}$$

For updating the Hessian matrix, one popular formula is the *BFGS formula*, which is

$$\boldsymbol{Y}_{k+1} = \boldsymbol{Y}_k - \frac{\boldsymbol{Y}_k \boldsymbol{s}_k \boldsymbol{s}_k^{\mathrm{T}} \boldsymbol{Y}_k}{\boldsymbol{s}_k^{\mathrm{T}} \boldsymbol{Y}_k \boldsymbol{s}_k} + \frac{\boldsymbol{g}_k \boldsymbol{g}_k^{\mathrm{T}}}{\boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{s}_k} \tag{2.39}$$

A good property of *BFGS update* is that it maintains the positive definite as long as the initial matrix $\boldsymbol{Y}_0$ is positive definite and $\boldsymbol{s}_k^{\mathrm{T}\boldsymbol{g}_k} > 0$.

In practical implementation, to reduce the computation cost involved in computing the inverse of Hessian in (2.34), the update can be done directly on the inverse Hessian. Let $\boldsymbol{H}_k := \boldsymbol{Y}_k^{-1}$, the update rule for $\boldsymbol{H}_k$ is:

$$\boldsymbol{H}_{k+1} = \left(\boldsymbol{I} - \rho_k \boldsymbol{s}_k \boldsymbol{g}_k^{\mathrm{T}}\right) \boldsymbol{H}_k \left(\boldsymbol{I} - \rho_k \boldsymbol{g}_k \boldsymbol{s}_k^{\mathrm{T}}\right) + \rho_k \boldsymbol{s}_k \boldsymbol{s}_k^{\mathrm{T}} \tag{2.40}$$

where

$$\rho_k = \frac{1}{\boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{s}_k} \tag{2.41}$$

Then the *Quasi-Newton* search direction with approximated Hessian matrix is:

$$\boldsymbol{p}_k^N = -\boldsymbol{H}_k \nabla f^0(\boldsymbol{x}_k) \tag{2.42}$$

2.5.4 Safe set formulation

During the optimization process, the safety requirement shall be fulfilled at each iteration. To ensure this requirement, a local safe set $\mathcal{S}_k \in \mathcal{D}$ is formulated at each iteration, so that the function evaluations inside this safe set do not violate constraints.

If a current safe point $\boldsymbol{x}_k$ is given, the safety requirement throughout the optimization process is equivalent to guarantee that moving from a safe point $\boldsymbol{x}_k$, the next point $\boldsymbol{x}_{k+1}$ does not violate constraints. That is, $f^i(\boldsymbol{x}_{k+1}) \leq 0$ for $\forall i = 1, \ldots, m$. A local safe set $\mathcal{S}_k^i$ for constraint $f^i(\boldsymbol{x})$ at point $\boldsymbol{x}_k$ is defined as the set in which all point $\boldsymbol{x}_{k+1}$ satisfy the safety constraints.

Given the function value at the current point $f^i(\boldsymbol{x}_k)$, with the smoothness assumption 1 about the unknown function, using the Taylor's theorem 1, the function value at the next point can be upper bounded by:

$$f^i(\boldsymbol{x}_{k+1}) \leq f^i(\boldsymbol{x}_k) + \nabla f^i(\boldsymbol{x}_k)^{\mathrm{T}}(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) + \frac{1}{2} M \|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|_2^2 \tag{2.43}$$

By forcing the upper bound less than zero, the safe set for $\boldsymbol{x}_{k+1}$ can be computed by:

$$\mathcal{S}_k^i := \{\boldsymbol{x} \in \mathbb{R}^d : f^i(\boldsymbol{x}_k) + \nabla f^i(\boldsymbol{x}_k)^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{x}_k) + \frac{1}{2} M \|\boldsymbol{x} - \boldsymbol{x}_k\|_2^2 \leq 0\} \tag{2.44}$$

Since in model-free optimization problem, the true gradient $\nabla f^i(\boldsymbol{x}_k)$ is not available, it is replaced by a gradient estimator $G^i(\boldsymbol{x}_k, \nu_k)$. To alleviate the effect of estimation deviation $\|\Delta_k^i\|_2$ on the safety properties of $\mathcal{S}_k^i$, some manipulations on the gradient estimator shall be made so that the computed safe set does not lost safety guarantee for the sake of estimation deviation. By the following manipulation, the safety of $\mathcal{S}_k^i$ is maintained in presence of estimation deviation.

Theorem 3. *Let $\boldsymbol{p}_k$ be the search direction, $G^i(\boldsymbol{x}_k, \nu_k)$ be the gradient estimator and $\|\Delta_k^i\|_2$ be the estimation deviation from $\nabla f^i(\boldsymbol{x}_k)$, then all points $\boldsymbol{x}_{k+1}$ satisfying the following constraints are feasible.*

$$f^i(\boldsymbol{x}_k) + \hat{G}^i(\boldsymbol{x}_k, \nu_k)^{\mathrm{T}}(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) + \frac{1}{2}M\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|_2^2 \leq 0 \tag{2.45}$$

where

$$\hat{G}^i(\boldsymbol{x}_k, \nu_k) = G^i(\boldsymbol{x}_k, \nu_k) + \frac{\sqrt{d}M\nu_k\|\boldsymbol{p}_k\|_2}{2\boldsymbol{e}^{\mathrm{T}}\boldsymbol{p}_k}\boldsymbol{e} \tag{2.46}$$

with a unit coordinate vector $\boldsymbol{e}$.

Proof. *Recall that $\boldsymbol{x}_{k+1}$ is feasible for constraint $f^i(\boldsymbol{x})$, if (2.43) holds. To prove Theorem 3, it is equivalent to prove that with (2.45), (2.43) always holds. Let α be the angle between $\nabla f^i(\boldsymbol{x}_k)$ and $\boldsymbol{p}_k$ and β be the angle between $G^i(\boldsymbol{x}_k, \nu_k)$ and $\boldsymbol{p}_k$. With (2.45), (2.43) holds if the following is true:*

$$\nabla f^i(\boldsymbol{x}_k)^{\mathrm{T}}(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) \leq \hat{G}^i(\boldsymbol{x}_k, \nu_k)^{\mathrm{T}}(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) \tag{2.47}$$

let $\boldsymbol{x}_{k+1} - \boldsymbol{x}_k = \alpha\boldsymbol{p}_k$, then (2.47) is equivalent to:

$$\alpha\nabla f^i(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{p}_k \leq \alpha\hat{G}^i(\boldsymbol{x}_k, \nu_k)^{\mathrm{T}}\boldsymbol{p}_k \tag{2.48}$$

$$\nabla f^i(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{p}_k \leq \hat{G}^i(\boldsymbol{x}_k, \nu_k)^{\mathrm{T}}\boldsymbol{p}_k \tag{2.49}$$

$$\nabla f^i(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{p}_k \leq G^i(\boldsymbol{x}_k, \nu_k)^{\mathrm{T}}\boldsymbol{p}_k + \frac{\sqrt{d}M\nu_k\|\boldsymbol{p}_k\|_2}{2\boldsymbol{e}^{\mathrm{T}}\boldsymbol{p}_k}\boldsymbol{e}^{\mathrm{T}}\boldsymbol{p}_k \tag{2.50}$$

$$\|\nabla f^i(\boldsymbol{x}_k)\|_2\|\boldsymbol{p}_k\|_2\cos\alpha \leq \|G^i(\boldsymbol{x}_k, \nu_k)\|_2\|\boldsymbol{p}_k\|_2\cos\beta + \frac{\sqrt{d}M\nu_k\|\boldsymbol{p}_k\|_2}{2} \tag{2.51}$$

$$\|\nabla f^i(\boldsymbol{x}_k)\|_2\cos\alpha - \|G^i(\boldsymbol{x}_k, \nu_k)\|_2\cos\beta \leq \frac{\sqrt{d}M\nu_k}{2} \tag{2.52}$$

Let ψ be the angle between $G^i(\boldsymbol{x}_k, \nu_k) - \nabla f^i(\boldsymbol{x}_k)$ and $-\boldsymbol{p}_k$. Since $G^i(\boldsymbol{x}_k, \nu_k)$ is either inside or on the surface of a ball $\mathcal{B}\left(\nabla f^i(\boldsymbol{x}_k), \|\Delta_k^i\|_2\right)$ centered at $\nabla f^i(\boldsymbol{x}_k)$ with radius $\|\Delta_k^i\|_2$ as shown in Figure 2.1, there is:

$$\left(\|\nabla f^i(\boldsymbol{x}_k)\|_2\cos\alpha - \|G^i(\boldsymbol{x}_k, \nu_k)\|_2\cos\beta\right)^2 = \left(\|\Delta_k^i\|_2\cos\psi\right)^2 \tag{2.53}$$

$$\left|\|\nabla f^i(\boldsymbol{x}_k)\|_2\cos\alpha - \|G^i(\boldsymbol{x}_k, \nu_k)\|_2\cos\beta\right| = \left|\|\Delta_k^i\|_2\cos\psi\right| \tag{2.54}$$

$$\|\nabla f^i(\boldsymbol{x}_k)\|_2\cos\alpha - \|G^i(\boldsymbol{x}_k, \nu_k)\|_2\cos\beta \leq \|\Delta_k^i\|_2 \leq \frac{\sqrt{d}M\nu_k}{2} \tag{2.55}$$

Thus (2.43) always holds, and $\boldsymbol{x}_{k+1}$ is feasible for constraint $f^i(\boldsymbol{x})$. □

With Theorem 3, a conservative safe set $\hat{\mathcal{S}}_k^i$ for constraint $f^i(\boldsymbol{x})$ is defined as:

$$\hat{\mathcal{S}}_k^i := \{\boldsymbol{x} \in \mathbb{R}^d : f^i(\boldsymbol{x}_k) + \hat{G}^i(\boldsymbol{x}_k, \nu_k)^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{x}_k) + \frac{1}{2}M\|\boldsymbol{x} - \boldsymbol{x}_k\|_2^2 \leq 0\} \tag{2.56}$$

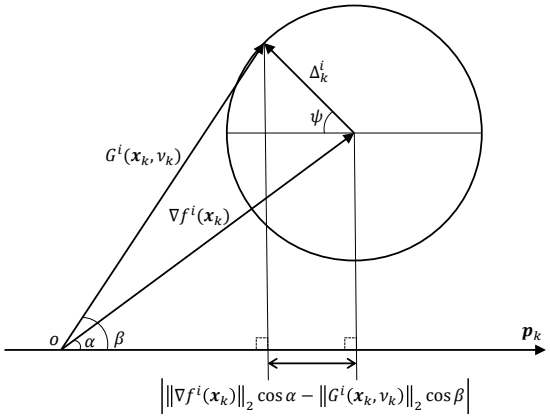


Figure 2.1: $G^i(\boldsymbol{x}_k, \nu_k)$ is inside or on the surface of a ball $\mathcal{B}\left(\nabla f^i(\boldsymbol{x}_k), \|\Delta_k^i\|_2\right)$

Lemma 3. *Safe set $\hat{\mathcal{S}}_k^i$ is a ball $\mathcal{B}$ with center $\hat{\mathcal{O}}_k^i$ and radius $\hat{\mathcal{R}}_k^i$:*

$$\hat{\mathcal{O}}_k^i = \boldsymbol{x}_k - \frac{1}{M}\hat{G}^i(\boldsymbol{x}_k, \nu_k) \tag{2.57}$$

$$\hat{\mathcal{R}}_k^i = \sqrt{\frac{1}{M}\left(\frac{\|\hat{G}^i(\boldsymbol{x}_k, \nu_k)\|_2^2}{M} - 2f^i(\boldsymbol{x}_k)\right)} \tag{2.58}$$

Proof. *Equation (2.45) can be rewritten as:*

$$\frac{1}{2}M\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|_2^2 + \hat{G}^i(\boldsymbol{x}_k, \nu_k)^{\mathrm{T}}(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) + f^i(\boldsymbol{x}_k) \leq 0 \tag{2.59}$$

$$\frac{1}{2}M\left(\|\boldsymbol{x}_{k+1}\|_2^2 + \|\boldsymbol{x}_k\|_2^2 - 2\boldsymbol{x}_k^{\mathrm{T}}\boldsymbol{x}_{k+1}\right) + \hat{G}^i(\boldsymbol{x}_k, \nu_k)^{\mathrm{T}}(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) + f^i(\boldsymbol{x}_k) \leq 0 \tag{2.60}$$

$$\frac{1}{2}M\|\boldsymbol{x}_{k+1}\|_2^2 + \left(\hat{G}^i(\boldsymbol{x}_k, \nu_k) - M\boldsymbol{x}_k\right)^{\mathrm{T}}\boldsymbol{x} + \frac{1}{2}M\|\boldsymbol{x}_k\|_2^2 - \hat{G}^i(\boldsymbol{x}_k, \nu_k)^{\mathrm{T}}\boldsymbol{x}_k + f^i(\boldsymbol{x}_k) \leq 0 \tag{2.61}$$

$$\|\boldsymbol{x}_{k+1}\|_2^2 + \underbrace{\frac{2}{M}\left(\hat{G}^i(\boldsymbol{x}_k, \nu_k) - M\boldsymbol{x}_k\right)^{\mathrm{T}}}_{P}\boldsymbol{x} + \underbrace{\|\boldsymbol{x}_k\|_2^2 - \frac{2}{M}\hat{G}^i(\boldsymbol{x}_k, \nu_k)^{\mathrm{T}}\boldsymbol{x}_k + \frac{2}{M}f^i(\boldsymbol{x}_k)}_{Q} \leq 0 \tag{2.62}$$

then, the center is

$$\hat{\mathcal{O}}_k^i = -\frac{1}{2}\cdot P \tag{2.63}$$

$$= -\frac{1}{2}\cdot\frac{2}{M}\left(\hat{G}^i(\boldsymbol{x}_k, \nu_k) - M\boldsymbol{x}_k\right) \tag{2.64}$$

$$= \boldsymbol{x}_k - \frac{1}{M}\hat{G}^i(\boldsymbol{x}_k, \nu_k) \tag{2.65}$$

and the radius is

$$\hat{\mathcal{R}}_k^i = \sqrt{\frac{1}{4}(\|P\|_2^2 - 4Q)} \tag{2.66}$$

$$= \sqrt{\frac{1}{4}\left[\left\|\frac{2}{M}(\hat{G}^i(\boldsymbol{x}_k, \nu_k) - M\boldsymbol{x}_k)\right\|_2^2 - 4\left[\|\boldsymbol{x}_k\|_2^2 - \frac{2}{M}\hat{G}^i(\boldsymbol{x}_k, \nu_k)^{\mathrm{T}}\boldsymbol{x}_k + \frac{2}{M}f^i(\boldsymbol{x}_k)\right]\right]} \tag{2.67}$$

$$= \sqrt{\frac{1}{M^2}\left(\|\hat{G}^i(\boldsymbol{x}_k, \nu_k)\|_2^2 + M^2\|\boldsymbol{x}_k\|_2^2\right) - \|\boldsymbol{x}_k\|_2^2 - \frac{2}{M}f^i(\boldsymbol{x}_k)} \tag{2.68}$$

$$= \sqrt{\frac{1}{M}\left(\frac{\|\hat{G}^i(\boldsymbol{x}_k, \nu_k)\|_2^2}{M} - 2f^i(\boldsymbol{x}_k)\right)} \tag{2.69}$$

□

Since the safety shall be satisfied for all constraints $f^i(\boldsymbol{x})$, $\forall i = 1, \ldots, m$, the overall safe set $\mathcal{S}_k$ at iteration k is the intersection of all $\hat{\mathcal{S}}_k^i$:

$$\mathcal{S}_k := \cap_{i=1}^m \hat{\mathcal{S}}_k^i \tag{2.70}$$

Using Lemma 3, safe set (2.70) is equivalent to:

$$\mathcal{S}_k := \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x} - \hat{\mathcal{O}}_k^i\|_2 \leq \hat{\mathcal{R}}_k^i,\ \forall i = 1, \ldots, m\} \tag{2.71}$$

With the computed safe set, the function evaluations can be made inside the safe set without violating constraints. The next step is to find a step length along the search direction inside the safe set, such that the objective function decreases rapidly.

2.5.5 Search direction projection

In constrained optimization problem, if the optimal point of the objective function is outside of the feasible set $\mathcal{D}$, the search direction shall be projected onto the normal vector of the surface spanned by gradients of all active constraints ($f^i(\boldsymbol{x}_k) = 0$). Since, if a current point is on a boundary of the feasible set, and the search direction is not parallel to the gradient of the active constraint and the angle between the search direction and the active constraint's gradient is less than $\frac{\pi}{2}$, then there exist a direction perpendicular to the constraint's gradient such that along this search direction, the objective can be reduced while not violating the constraint.

Let us assume that the search direction $\boldsymbol{p}_k$ is the negative gradient of the objective function $-\nabla f^0(\boldsymbol{x}_k)$, an illustration of the search direction projection can be seen in Figure 2.2.

Lemma 4. *A point is a KKT point of the constrained optimization problem 2, if at this point, the gradient of the objective function is a linear combination of gradients of all active constraints with non-negative coefficients.*

Proof. *Let $\mathcal{A}$ be the set of indices of all active constraints at point $\boldsymbol{x}_k$, set all Lagrangian multipliers for inactive constraints to zeros:*

$$f^i(\boldsymbol{x}_k) < 0, \ i \notin \mathcal{A} \tag{2.72}$$

$$\lambda_k^i = 0, \ i \notin \mathcal{A} \tag{2.73}$$

If there exists a set of non-negative multipliers for active constraints such that the following holds:

$$-\nabla f^0(\boldsymbol{x}_k) = \sum_{i \in \mathcal{A}} \lambda_k^i \nabla f^i(\boldsymbol{x}_k) \tag{2.74}$$

$$f^i(\boldsymbol{x}_k) = 0, \ i \in \mathcal{A} \tag{2.75}$$

$$\lambda_k^i \geq 0, \ i \in \mathcal{A} \tag{2.76}$$

Then, the derivative of Lagrangian $\mathcal{L}(\boldsymbol{x}_k, \boldsymbol{\lambda}_k)$ 2.11 can be written as:

$$\nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}_k, \boldsymbol{\lambda}_k) = \nabla f^0(\boldsymbol{x}_k) + \sum_{i \in \mathcal{A}} \lambda_k^i \nabla f^i(\boldsymbol{x}_k) + \sum_{i \notin \mathcal{A}} \lambda_k^i \nabla f^i(\boldsymbol{x}_k) = 0 \tag{2.77}$$

and there is:

$$f^i(\boldsymbol{x}_k) \leq 0, \ i = 1, \ldots, m \tag{2.78}$$

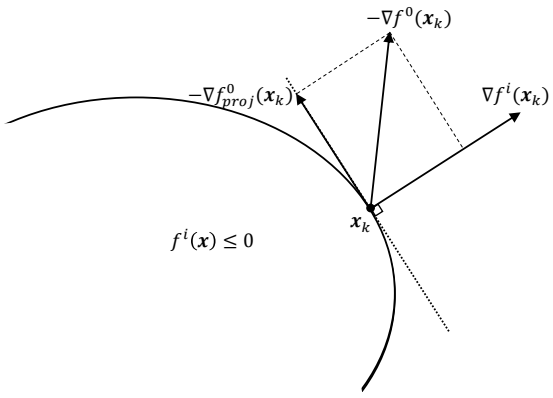


Figure 2.2: Search direction $\boldsymbol{p}_k = -\nabla f^0(\boldsymbol{x}_k)$ projected onto the normal vector of constraint's gradient $\nabla f^i(\boldsymbol{x}_k)$

$$\lambda_k^i \geq 0,\ i = 1, \ldots, m \tag{2.79}$$
$$\lambda_k^i f^i(\boldsymbol{x}_k) = 0,\ i = 1, \ldots, m \tag{2.80}$$

Thus, the KKT condition 2 is satisfied and the point $\boldsymbol{x}_k$ is a KKT point. □

Geometrically, (2.74) means that the negative gradient of objective function is inside a convex cone spanned by all active constraint's gradients.
The search direction projection $\boldsymbol{p}_k^{proj}$ can be found by the following:

Lemma 5. *Let $\boldsymbol{C}_k \in \mathbb{R}^{d \times |\mathcal{A}|}$ be the the matrix with columns of gradients of all active constraints $\nabla f^i(\boldsymbol{x}_k)$, $i \in \mathcal{A}$, the minimal angle between $-\nabla f^0(\boldsymbol{x}_k)$ and $f^i(\boldsymbol{x}_k)$ is less than $\frac{\pi}{2}$, $\boldsymbol{\lambda}_{sol} \in \mathbb{R}^{|\mathcal{A}|}$ be the solution vector of the following non-negative linear least-squares problem:*

Problem 3. ***Non-Negative Linear Least Squares Problem (Nonneglsp)***

$$\underset{\boldsymbol{\lambda} \in \mathbb{R}^{|\mathcal{A}|}}{\text{minimize}} \quad \|\boldsymbol{C}_k \boldsymbol{\lambda} - \boldsymbol{p}_k\|_2^2 \tag{2.81}$$
$$\text{subject to} \quad \lambda^i \geq 0,\ i \in \mathcal{A} \tag{2.82}$$

Direction $\boldsymbol{p}_k^{proj}$ is a descent direction and all active constraints do not increase along this direction, if $\boldsymbol{x}_k$ is not a KKT point and:

$$\boldsymbol{p}_k^{proj} = \boldsymbol{p}_k - \boldsymbol{p}_k' \tag{2.83}$$

where $\boldsymbol{p}_k$ is the negative gradient of the objective function:

$$\boldsymbol{p}_k = -\nabla f^0(\boldsymbol{x}_k) \tag{2.84}$$

and $\boldsymbol{p}_k'$ is:

$$\boldsymbol{p}_k' = \boldsymbol{C}_k \boldsymbol{\lambda}_{sol} \tag{2.85}$$

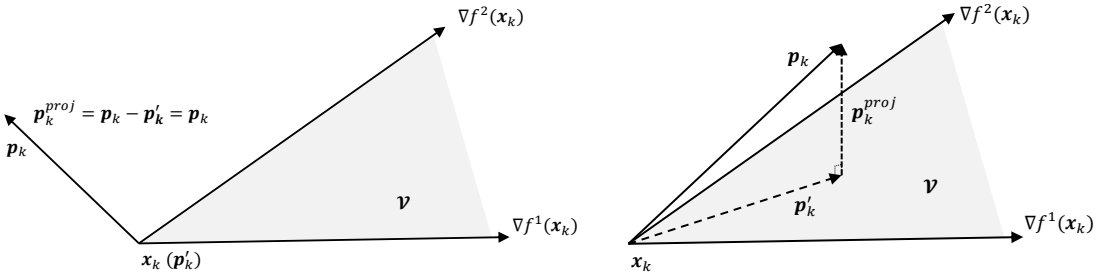
Proof. *Let J_{min} be the minimum of (2.81) at point $\boldsymbol{x}_k$. If $J_{min} = 0$, then $\boldsymbol{\lambda}_{sol}$ is a non-negative multiplier vector such that the following holds:*

$$-\nabla f^0(\boldsymbol{x}_k) = \boldsymbol{C}_k \boldsymbol{\lambda}_{sol} = \sum_{i \in \mathcal{A}} \lambda_{sol}^i \nabla f^i(\boldsymbol{x}_k) \tag{2.86}$$

According to Lemma 4, the point $\boldsymbol{x}_k$ is a KKT point.

If $J_{min} > 0$, then $-\nabla f^0(\boldsymbol{x}_k)$ can not be exactly expressed by a linear combination of gradients of all active constraints, thus, $\boldsymbol{x}_k$ is not a KKT point.

Let φ be the angle between $\boldsymbol{p}_k$ and an arbitrary vector $\boldsymbol{p}_k^a$ in the convex



(a) $\boldsymbol{p}_k^{proj}$ when $\varphi_{min} \geq \frac{\pi}{2}$ (b) $\boldsymbol{p}_k^{proj}$ when $\varphi_{min} < \frac{\pi}{2}$

Figure 2.3: Search direction projection $\boldsymbol{p}_k^{proj}$ in $\mathbb{R}^3$

cone $\mathcal{V}$ *spanned by* $\nabla f^i(\boldsymbol{x}_k),\ i \in \mathcal{A}$, γ *be the angle between any two vectors of* $\nabla f^i(\boldsymbol{x}_k),\ i \in \mathcal{A}$*:*

$$\varphi = \measuredangle\left(\boldsymbol{p}_k,\ \boldsymbol{p}_k^a\right),\ \forall \boldsymbol{p}_k^a \in \mathcal{V} \tag{2.87}$$

$$\gamma = \measuredangle\left[\nabla f^i(\boldsymbol{x}_k),\ \nabla f^j(\boldsymbol{x}_k)\right],\ i, j \in \mathcal{A} \tag{2.88}$$

One can see that:

$$J_{min} = \min \|\boldsymbol{C}_k \boldsymbol{\lambda}_{sol} - \left(-\nabla f^0(\boldsymbol{x}_k)\right)\|_2^2 = \min \|\boldsymbol{p}'_k - \boldsymbol{p}_k\|_2^2 \tag{2.89}$$

which means that $\boldsymbol{p}'_k$ *is the best approximation of* $\boldsymbol{p}_k$ *among all possible* $\boldsymbol{p}_k^a$ *in the convex cone* $\mathcal{V}$*. Geometrically, it is equivalent to that the distance between* $\boldsymbol{p}'_k$ *and* $\boldsymbol{p}_k$ *is minimal among all possible distances between* $\boldsymbol{p}_k$ *and* $\boldsymbol{p}_k^a$*, and thus* $\boldsymbol{p}_k^{proj} = \boldsymbol{p}_k - \boldsymbol{p}'_k$ *is perpendicular to* $\boldsymbol{p}'_k$ *and the minimal possible angle* φ_{min} *is the angle between* $\boldsymbol{p}_k$ *and* $\boldsymbol{p}'_k$*.*

$$\varphi_{min} = \measuredangle(\boldsymbol{p}_k,\ \boldsymbol{p}'_k) \tag{2.90}$$

If $\varphi_{min} \geq \frac{\pi}{2}$*, then there is* $\boldsymbol{\lambda}_{sol} = \boldsymbol{0}$ *and* $\|\boldsymbol{p}'_k\|_2 = 0$*, which means that* $\boldsymbol{p}'_k$ *is a point located at* $\boldsymbol{x}_k$*, and thus,* $\boldsymbol{p}_k^{proj} = \boldsymbol{p}_k - \boldsymbol{p}'_k = \boldsymbol{p}_k$*. Since* $\boldsymbol{p}_k = -\nabla f^0(\boldsymbol{x}_k)$*, thus* $\boldsymbol{p}_k^{proj}$ *is a descent direction.*

if $\varphi_{min} < \frac{\pi}{2}$*, then there is:*

$$\measuredangle(\boldsymbol{p}_k,\ \boldsymbol{p}_k^{proj}) = \frac{\pi}{2} - \measuredangle(\boldsymbol{p}_k,\ \boldsymbol{p}'_k) = \frac{\pi}{2} - \varphi_{min} \leq \frac{\pi}{2} \tag{2.91}$$

Thus, $\boldsymbol{p}_k^{proj}$ *is a descent direction. An illustration of search direction projection in* $\mathbb{R}^3$ *is shown in Figure 2.3.*

As one knows that in a convex cone $\mathcal{V}$*, the maximum possible angle* γ_{max} *between any two vectors of* $\nabla f^i(\boldsymbol{x}_k),\ i \in \mathcal{A}$ *satisfies* $\gamma_{max} \leq \pi$ *(otherwise* $\mathcal{V}$ *is not convex), hence, the angle between* $\boldsymbol{p}_k^{proj}$ *and any vector* $\boldsymbol{p}_k^a$ *in* $\mathcal{V}$ *satisfies:*

$$\frac{\pi}{2} \leq \measuredangle\left(\boldsymbol{p}_k^{proj}, \boldsymbol{p}_k^a\right) \leq \frac{\pi}{2} + \gamma_{max} \leq \frac{3\pi}{2} \tag{2.92}$$

Thus, along the direction $\boldsymbol{p}_k^{proj}$*, all active constraints do not increase.* □

Problem 3 can be solved using non-negative least-square solvers such as the *lsqnonneg* function in numerical analysis software MATLAB.

The search direction projection is computed until J_{min} converges to zero. The decision point in this case is a KKT point of the constrained optimization problem 2. In practice, this condition is satisfied if J_{min} is less than some tolerance value ϵ, e.g. $\epsilon = 10^{-10}$.

2.5.6 Step length computation

The choice of step length shall have the following properties: first, it provides a sufficient decrease in the objective function; second, the function evaluation at the next step shall not violate the safety constraints.

Ideally, with the selected step length, the next function evaluation shall have the minimum possible value along this search direction, while satisfies safety constraints. Denote by α_k the step length at iteration k, finding a step length with above properties is equivalent to solve the following optimization problem:

$$\begin{aligned} \underset{\alpha}{\text{minimize}} \quad & f^0(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k) \\ \text{subject to} \quad & \boldsymbol{x}_k + \alpha \boldsymbol{p}_k \in \mathcal{S}_k \\ & \alpha_k > 0 \end{aligned} \tag{2.93}$$

Solving this problem aims to find a step length that minimizes the objective function at the next iteration while satisfying safety constraints. However, it is generally expensive to find a global minimizer of this problem, as there may be many local optimums along the search direction. A practical way is to find an inexact solution of the problem, with which the objective decreases not by the maximum possible value, instead, by only a good enough amount, and then iteatively improve the solution. An inexact solution of the step length can be found using the *line search* algorithm. In the algorithm, it tries a set of candidate step lengths and selects the one that provides enough decrease in the objective function.

A simple condition for selection of candidate step length is:

$$f^0(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k) < f^0(\boldsymbol{x}_k) \tag{2.94}$$

which means that the step length is acceptable as long as the objective at the next iteration is less than the current. However, this condition may lead to a poor convergence performance, as any little reduction in the objective function is accepted. In order to provide a better convergence result, a *sufficient decrease condition* is applied. By this condition, only the step length with which the objective decreases by enough amount are selected. The *Armijo condition* states such a measurement, which is:

$$f^0(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k) < f^0(\boldsymbol{x}_k) + c\alpha_k \nabla f^0(\boldsymbol{x}_k)^{\mathrm{T}} \boldsymbol{p}_k \tag{2.95}$$

where $c \in (0,1)$ is a constant that measures the decrease level. The *Armijo condition* implies that with the step length α_k along the search direction $\boldsymbol{p}_k$, the decrease in the objective function are larger than $\|c\alpha_k \nabla f^0(\boldsymbol{x}_k)\boldsymbol{p}_k\|_2$.

To illustrate the effect of constant c, let $l(\alpha)$, $h(\alpha)$ be:

$$l(\alpha) = f^0(\boldsymbol{x}_k) + c\alpha_k \nabla f^0(\boldsymbol{x}_k)^{\mathrm{T}} \boldsymbol{p}_k \tag{2.96}$$

$$h(\alpha) = f^0(\boldsymbol{x}_k) + \alpha_k \nabla f^0(\boldsymbol{x}_k)^{\mathrm{T}} \boldsymbol{p}_k \tag{2.97}$$

the *Armijo condition* can be interpreted by Figure 2.4. It can be seen that with

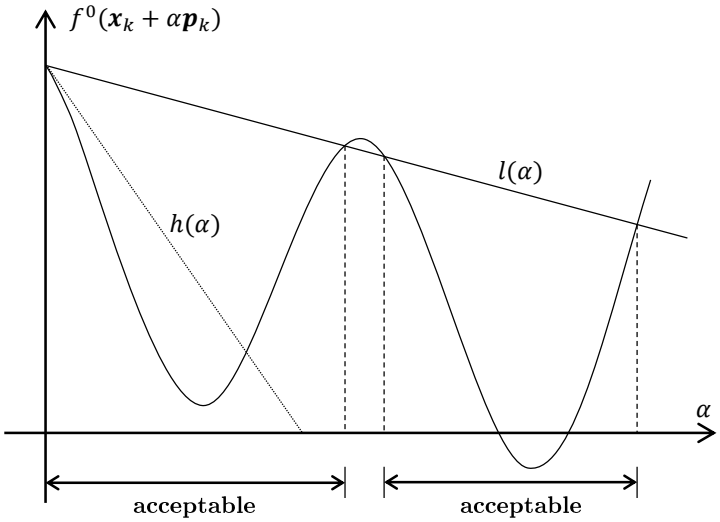


Figure 2.4: Acceptable step length

different constant c, the range of acceptable step lengths varies. In practice, c is often chosen to be a small value to incorporate a larger range. In model-free optimization problem, the exact gradient $\nabla f^0(\boldsymbol{x}_k)$ is replaced by a gradient estimator $G^0(\boldsymbol{x}_k, \nu_k)$.

In addition to the sufficient decrease requirement, in safe optimization problem, the safety requirement shall be satisfied. As the line search approach tries a sequence of candidate step lengths, a trial of too large step length may lead to violation of the safety constraints. To overcome this problem, an upper bound on the possible candidate step lengths are defined, within which all candidate step lengths satisfy the safety constraints. The upper bound of candidate step lengths is such that all trial points used when searching for the step length are inside the safe set $\mathcal{S}_k$. Using (2.71), the upper bound of candidate step lengths $\hat{\alpha}_k$ along the search direction $\boldsymbol{p}_k$ is defined as:

$$\hat{\alpha}_k = \max\{\alpha \in \mathbb{R} : \|\boldsymbol{x}_k + \alpha \boldsymbol{p}_k - \hat{\mathcal{O}}_k^i\|_2 < \hat{\mathcal{R}}_k^i, \forall i = 1, \ldots, m, \alpha > 0\} \tag{2.98}$$

The step length upper bound is illustrated in Figure 2.5.

Besides the upper bound on candidate step length, during testing step lengths,

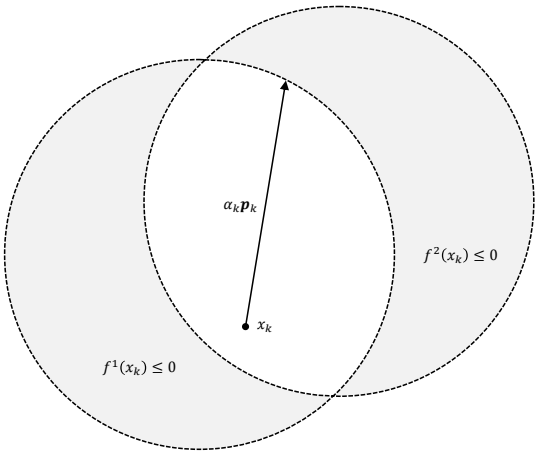


Figure 2.5: Step length upper bound

an additional safety condition shall be satisfied by acceptable step length. Since in model-free optimization, the gradient estimators and function measurements involve uncertainties, the safety constraints $f^i(\boldsymbol{x}) \leq 0$ may be violated if it is extremely close to a boundary. Instead, a conservative safety constraints are applied. It is defined as:

$$-f^i(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k) \geq h, \ \forall i = 1, \ldots, m \tag{2.99}$$

where $h > 0$ is a safety threshold. With the conservative safety constraints, the decision points will keep a distance of h to the actual boundary to increase robustness.

In searching for the acceptable step length, the conservative safety constraints shall be satisfied in addition to the requirement of sufficient decrease.

With above consideration, the safe step length selection algorithm is formulated in Algorithm 4

Algorithm 4 Safe step length selection for exact measurements (e-StepLength)

Input: Search direction $\boldsymbol{p}_k$, gradient estimator for objective $G^0(\boldsymbol{x}_k, \nu_k)$, gradient estimator for constraints $\hat{G}^i(\boldsymbol{x}_k, \nu_k)$, centers and radii $\hat{\mathcal{O}}_k^i$, $\hat{\mathcal{R}}_k^i$, $i = 1, \ldots, m$, $\hat{\alpha}_k > 0$, $h > 0$, $\rho \in (0, 1)$, $c \in (0, 1)$;
while $\min\limits_{i=1,\ldots,m} \left\{ \hat{\mathcal{R}}_k^i - \| (\boldsymbol{x}_k + \hat{\alpha}_k \boldsymbol{p}_k) - \hat{\mathcal{O}}_k^i \|_2 \right\} \leq 0$ **do**
    $\hat{\alpha}_k = \rho \hat{\alpha}_k$;
end while
Set $\alpha_k \leftarrow \hat{\alpha}_k$;
Evaluate $f^i(\boldsymbol{x}_k)$ and $f^i(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k)$, $i = 0, \ldots, m$;
while $f^0(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k) \geq f^0(\boldsymbol{x}_k) + c\alpha_k G^0(\boldsymbol{x}_k, \nu_k)^{\mathrm{T}} \boldsymbol{p}_k$ **or** $\min\limits_{i=1,\ldots,m} \{-f^i(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k)\} < h$ **do**
    $\alpha_k = \rho \alpha_k$;
    Update $f^i(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k)$, $i = 0, \ldots, m$;
end while
Return: α_k

2.5.7 Safe-line-search algorithm (e-SLS) formulation

With the determined search direction $\boldsymbol{p}_k$, the safe set $\mathcal{S}_k$ and the acceptable step length that provides sufficient decrease in the objective function and satisfies the safety constraints. The safe-line-search algorithm for exact measurement (e-SLS) can be formulated. An outline of the algorithm is summarized as:

1. Compute the gradient estimator $\hat{G}^i(\boldsymbol{x}_k)$ for $f^i(\boldsymbol{x}_k),\ i = 0, \ldots, m$;
2. Determine the search direction $\boldsymbol{p}_k$;
3. Compute the search direction projection $\boldsymbol{p}_k^{proj}$ if $\min\limits_{i=1,\ldots,m} -f^i(x_k)$ is below a safety threshold h;
4. Find the local safe set $\mathcal{S}_k$;
5. Compute a safety-guaranteed step length α_k;
6. Set the next point as $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$;
7. Evaluate functions at $f^i(\boldsymbol{x}_{k+1}),\ i = 0, \ldots, m$;
8. Check the convergence condition.

The complete algorithm is shown in Algorithm 5.

Algorithm 5 Safe-line-search algorithm for exact measurements (e-SLS)

Input: $\boldsymbol{x}_0 \in \mathcal{D}$, $f^i(\boldsymbol{x}_0), i = 0, \ldots, m$, L, $M > 0$, $d > 0$, $\mu > 0$, $h > 0$, $\epsilon > 0$, $\rho \in (0, 1)$, $c \in (0, 1)$;
for $k = 0, 1, \ldots$ **do**
  Take $\nu_k = \min\left\{\frac{2\mu}{\sqrt{d}M}, \frac{\min\limits_{i=1,\ldots,m}\{-f^i(\boldsymbol{x}_k)\}}{2L}\right\}$;
  For each $f^i(\boldsymbol{x})$, $i = 0, \ldots, m$, make evaluations at points $\boldsymbol{x}_k + \nu_k \boldsymbol{e}_j$, $j = 1, \ldots, d$;
  Compute the gradient estimators $G^i(\boldsymbol{x}_k, \nu_k)$ of $\nabla f^i(\boldsymbol{x}_k)$, $i = 0, \ldots, m$ using (2.17);
  Set $\boldsymbol{p}_k = -G^0(\boldsymbol{x}_k, \nu_k)$ **or** *Quasi-Newton* direction (2.42);
  Compute $\hat{G}^i(\boldsymbol{x}_k, \nu_k)$, $i = 1, \ldots, m$ (2.46);
  Compute the centers $\hat{\mathcal{O}}_k^i$ and radii $\hat{\mathcal{R}}_k^i$ of safe set $\hat{\mathcal{S}}_k^i$, $i = 1, \ldots, m$ (2.57), (2.58);
  if $\min\{-f^i(\boldsymbol{x}_k)\} \leq h$ **then**
    $\mathcal{A} := \{i : -f^i(\boldsymbol{x}_k)\} \leq h\}$;
    $\boldsymbol{C}_k = \left[\hat{G}^i(\boldsymbol{x}_k, \nu_k), \ldots, \hat{G}^j(\boldsymbol{x}_k, \nu_k)\right]$, $i, \ldots, j \in \mathcal{A}$;
    $\boldsymbol{\lambda}_{sol} \leftarrow$ Solve Non-negative LS problem 3;
    $\boldsymbol{p}_k^{proj} = \boldsymbol{p}_k - \boldsymbol{C}_k \boldsymbol{\lambda}_{sol}$;
    $\boldsymbol{p}_k = \boldsymbol{p}_k^{proj}$;
  end if
  Find $\alpha_k \leftarrow$ e-StepLength 4;
  Set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$;
  Evaluate $f^i(\boldsymbol{x}_{k+1})$, $i = 0, \ldots, m$;
  if $\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|_2 \leq \epsilon$ **then**
    $\boldsymbol{x}_* = \boldsymbol{x}_k$;
    $\boldsymbol{\lambda}_* = \boldsymbol{0}$;
    if $\mathcal{A} \neq \emptyset$ **then**
      $\lambda_*^i = \lambda_{sol}^i$, $i \in \mathcal{A}$;
    end if
    Algorithm Stop;
  end if
end for

2.6 Safe-line-search algorithm for noisy measurements (n-SLS)

2.6.1 Influence of measurement noises

One of the main challenges of model-free optimization methods is the measurement noises. Unlike function evaluations in simulation, in practical implementation, the measurements of function values are always corrupted with noises. Those noises impose two main limitations on the safe derivative-free optimization algorithm:

1. It reduces the accuracy of safety constraints function measurements;
2. It significantly enlarges the estimation error of gradient estimators.

In the following sections, the modification of the algorithm for exact measurements for handling measurement noises are discussed. Firstly, some notations and assumption are made.

Denote by ξ the noise variable, and assume that ξ is independently and identically distributed (i.i.d) zero-mean-σ-sub-Gaussian 3. A sub-Gaussian variable X denoted by $X \sim \text{subG}(\sigma^2)$, has the following property:

Theorem 4. *Let $X \sim subG(\sigma^2)$. Then for any $b > 0$, there is:*

$$\mathbb{P}[X > b] \leq \exp\left(-\frac{b^2}{2\sigma^2}\right) \tag{2.100}$$

$$\mathbb{P}[X < -b] \leq \exp\left(-\frac{b^2}{2\sigma^2}\right) \tag{2.101}$$

If $X_1, \ldots, X_n$ are i.i.d subG(σ^2), then the average $\bar{X} = \frac{1}{n}\sum_{i=1}^{n}$ is:

$$\bar{X} \sim \text{subG}(\frac{\sigma^2}{n}) \tag{2.102}$$

Using (2.100) and (2.101), $\bar{X}$ satisfies:

$$\mathbb{P}[\bar{X} > b] \leq \exp\left(-\frac{nb^2}{2\sigma^2}\right) \tag{2.103}$$

$$\mathbb{P}[\bar{X} < -b] \leq \exp\left(-\frac{nb^2}{2\sigma^2}\right) \tag{2.104}$$

With the assumption on measurement noise as sub-Gaussian random variable, using its properties, the modification of safe-line-search algorithm for noisy measurements is stated below.

2.6.2 Modification of e-SLS algorithm for noisy measurements

The modification for noisy measurements is based on the idea that replacing variables of single measurements by a sample-average ones can reduce the variances of uncertain variables. The modification for individual component in the Algorithm 5 is derived below.

The first modification is on the Gradient estimator. Let n_k be the number of repeated measurements, ξ^i_{0l} be the measurement noise of l-th measurement of constraint $f^i(\boldsymbol{x}_k)$, $\tilde{f}^i(\boldsymbol{x}_k;\xi^i_{0l})$ be the l-th noise-corrupted measurement of $f^i(\boldsymbol{x}_k)$:

$$\tilde{f}^i(\boldsymbol{x}_k;\xi^i_{0l}) = f^i(\boldsymbol{x}_k) + \xi^i_{0l} \tag{2.105}$$

and $\tilde{f}^i(\boldsymbol{x}_k+\nu_k\boldsymbol{e}_j;\xi^i_{jl})$ be the l-th measurement of $f^i(\boldsymbol{x}_k+\nu_k\boldsymbol{e}_j)$

$$\tilde{f}^i(\boldsymbol{x}_k+\nu_k\boldsymbol{e}_j;\xi^i_{jl}) = f^i(\boldsymbol{x}_k+\nu_k\boldsymbol{e}_j) + \xi^i_{jl} \tag{2.106}$$

Let $\bar{f}^i(\boldsymbol{x}_k;\xi^i_0)$ and $\bar{f}^i(\boldsymbol{x}_k+\nu_k\boldsymbol{e}_j;\xi^i_j)$ be the mean values of $\tilde{f}^i(\boldsymbol{x}_k;\xi^i_{0l})$ and $\tilde{f}^i(\boldsymbol{x}_k+\nu_k\boldsymbol{e}_j;\xi^i_{jl})$.

$$\bar{f}^i(\boldsymbol{x}_k;\xi^i_0) = \frac{\sum_{l=1}^{n_k}\tilde{f}^i(\boldsymbol{x}_k;\xi^i_{0l})}{n_k} \tag{2.107}$$

$$\bar{f}^i(\boldsymbol{x}_k+\nu_k\boldsymbol{e}_j;\xi^i_j) = \frac{\sum_{l=1}^{n_k}\tilde{f}^i(\boldsymbol{x}_k+\nu_k\boldsymbol{e}_j;\xi^i_{jl})}{n_k} \tag{2.108}$$

A sample-average gradient estimator $\bar{G}^i(\boldsymbol{x}_k,\nu_k;\xi^i)$ of $\nabla f^i(\boldsymbol{x}_k)$ is computed by:

$$\bar{G}^i(\boldsymbol{x}_k,\nu_k;\xi^i) = \sum_{j=1}^{d}\frac{\bar{f}^i(\boldsymbol{x}_k+\nu_k\boldsymbol{e}_j;\xi^i_j) - \bar{f}^i(\boldsymbol{x}_k;\xi^i_0)}{\nu_k}\boldsymbol{e}_j \tag{2.109}$$

In case of noisy measurement, the estimation error becomes larger than the case of exact measurements, and it comes from both the finite difference approximation, and the measurement noises. According to the *Law of large numbers*, the estimation error from noisy measurements can be reduced by increasing the number of measurements n_k, since with larger n_k, the average tends to be closer to the expected value.

Denote the gradient estimation error with noisy measurements by Δ^i_{k,ξ^i}:

$$\Delta^i_{k,\xi^i} = \bar{G}^i(\boldsymbol{x}_k,\nu_k;\xi^i) - \nabla f^i(\boldsymbol{x}_k) \tag{2.110}$$

The upper bound of Δ^i_{k,ξ^i} can be determined by the following.

Theorem 5. *The gradient estimation error $\|\Delta^i_{k,\xi^i}\|_2$ with zero-mean-σ-sub-Gaussian noise is upper bounded with high probability $1-\delta$ by:*

$$\mathbb{P}\left\{\|\Delta^i_{k,\xi^i}\|_2 \le \sqrt{\frac{dM^2\nu_k^2}{4} - \frac{4d\sigma^2\ln\delta}{\nu_k^2 n_k}}\right\} \ge 1-\delta \tag{2.111}$$

To balance the two terms inside the square root, let $n_k = -\frac{16\sigma^2 \ln \delta}{3\nu_k^4 M^2}$ *, then*

$$\mathbb{P}\left\{\|\Delta^i_{k,\xi^i}\|_2 \leq \sqrt{d} M \nu_k\right\} \geq 1 - \delta \tag{2.112}$$

Proof. *Let* ξ^i_j *be the mean value of* ξ^i_{jl}*:*

$$\xi^i_j = \sum_{l=1}^{n_k} \xi^i_{jl}, \ \ j = 0, \ldots, d \tag{2.113}$$

Recall the M-smooth property 2 of $f^i(\boldsymbol{x}_k)$*:*

$$f^i(\boldsymbol{x}_k + \nu_k \boldsymbol{e}_j) \leq f^i(\boldsymbol{x}_k) + \nabla f^i(\boldsymbol{x}_k)^{\mathrm{T}}(\nu_k \boldsymbol{e}_j) + \frac{1}{2} M \|\nu_k \boldsymbol{e}_j\|_2^2 \tag{2.114}$$

Using (2.109) and (2.110), the estimation error can be written as.

$$\|\Delta^i_{k,\xi^i}\|_2 = \sqrt{\sum_{j=1}^{d} \left[\frac{f^i(\boldsymbol{x}_k + \nu_k \boldsymbol{e}_j) - f^i(\boldsymbol{x}_k) + \xi^i_j - \xi^i_0}{\nu_k} - \nabla f^i(\boldsymbol{x}_k)^{\mathrm{T}} \boldsymbol{e}_j\right]^2 \|\boldsymbol{e}_j\|_2^2} \tag{2.115}$$

$$\leq \sqrt{\sum_{j=1}^{d} \left[\frac{f^i(\boldsymbol{x}_k + \nu_k \boldsymbol{e}_j) - f^i(\boldsymbol{x}_k) - \nabla f^i(\boldsymbol{x}_k)^{\mathrm{T}}(\nu_k \boldsymbol{e}_j)}{\nu_k}\right]^2 + \frac{1}{\nu_k^2} \sum_{j=1}^{d} \left(\xi^i_j - \xi^i_0\right)^2} \tag{2.116}$$

$$\leq \sqrt{\sum_{j=1}^{d} \left[\frac{M\|\nu_k \boldsymbol{e}_j\|_2^2}{2\nu_k}\right]^2 + \frac{1}{\nu_k^2} \sum_{j=1}^{d} \left(\xi^i_j - \xi^i_0\right)^2} \tag{2.117}$$

Let $\tilde{\xi}^i = \xi^i_j - \xi^i_0$*, since* $\xi^i_{jl} \sim \mathrm{subG}(\sigma^2)$*,* $j = 0, \ldots, d$*, according to (2.102), there is:*

$$\xi^i_j \sim \mathrm{subG}(\frac{\sigma^2}{n_k}), \ \ j = 0, \ldots, d \tag{2.118}$$

and

$$\tilde{\xi}^i \sim \mathrm{subG}(\frac{2\sigma^2}{n_k}) \tag{2.119}$$

then using 2.103 there is:

$$\mathbb{P}\left\{\bar{\xi}^i > b\right\} \leq \exp\left(-\frac{n_k b^2}{4\sigma^2}\right) = \delta \tag{2.120}$$

$$b^2 = -\frac{4\sigma^2 \ln \delta}{n_k} \tag{2.121}$$

thus:

$$\mathbb{P}\left\{\bar{\xi}^{i2} < -\frac{4\sigma^2 \ln \delta}{n_k}\right\} \geq 1-\delta \tag{2.122}$$

then (2.117) is equivalent to:

$$\mathbb{P}\left\{\|\Delta^i_{k,\xi^i}\|_2 \leq \sqrt{\frac{dM^2\nu_k^2}{4} - \frac{4d\sigma^2 \ln \delta}{\nu_k^2 n_k}}\right\} \geq 1-\delta \tag{2.123}$$

When $n_k = -\frac{16\sigma^2 \ln \delta}{3\nu_k^4 M^2}$, *there is:*

$$\mathbb{P}\left\{\|\Delta^i_{k,\xi^i}\|_2 \leq \sqrt{d}M\nu_k\right\} \geq 1-\delta \tag{2.124}$$

□

Thus, in case of noisy measurements, the gradient estimator $G^i(\boldsymbol{x}_k, \nu_k)$ is replaced by $\bar{G}^i(\boldsymbol{x}_k, \nu_k; \xi^i)$ (2.109) and the estimation deviation becomes Δ^i_{k,ξ^i} (2.111).

The next component to be modified is the safe set $\mathcal{S}_k$ at each iteration. With the upper bound of the estimation deviation $\|\Delta^i_{k,\xi^i}\|_2$, a conservative gradient estimator can be made by the following:

Theorem 6. *Let* $\boldsymbol{p}_k$ *be the descent direction with unit length,* $G^i(\boldsymbol{x}_k, \nu_k; \xi^i)$ *be the gradient estimator from noisy measurements,* $\|\Delta^i_{k,\xi^i}\|_2$ *be the estimation error and* $n_k = -\frac{16\sigma^2 \ln \delta}{3\nu_k^4 M^2}$, *then with high probability* $1-\delta$, *all points* $\boldsymbol{x}_{k+1}$ *satisfying the following constraints are feasible:*

$$\bar{f}^i(\boldsymbol{x}_k; \xi_0^i) + \hat{G}^i(\boldsymbol{x}_k, \nu_k; \xi^i)^{\mathrm{T}}(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) + \frac{1}{2}M\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|_2^2 \leq 0 \tag{2.125}$$

where

$$\hat{G}^i(\boldsymbol{x}_k, \nu_k; \xi^i) = \bar{G}^i(\boldsymbol{x}_k, \nu_k; \xi^i) + \sqrt{\frac{dM^2\nu_k^2}{4} - \frac{4d\sigma^2 \ln \delta}{\nu_k^2 n_k}}\frac{\|\boldsymbol{p}_k\|_2 \boldsymbol{e}}{\boldsymbol{e}^{\mathrm{T}}\boldsymbol{p}_k} \tag{2.126}$$

with $n_k = -\frac{16\sigma^2 \ln \delta}{3\nu_k^4 M^2}$, *then:*

$$\hat{G}^i(\boldsymbol{x}_k, \nu_k; \xi^i) = \bar{G}^i(\boldsymbol{x}_k, \nu_k; \xi^i) + \frac{\sqrt{d}M\nu_k\|\boldsymbol{p}_k\|_2}{\boldsymbol{e}^{\mathrm{T}}\boldsymbol{p}_k}\boldsymbol{e} \tag{2.127}$$

Proof. *For simplicity in this proof, let* $n_k = -\frac{16\sigma^2 \ln \delta}{3\nu_k^4 M^2}$. *Similar to Proof 2.5.4, if* $\boldsymbol{x}_{k+1}$ *is feasible, there is*

$$\bar{f}^i(\boldsymbol{x}_k; \xi_0^i) + \nabla f^i(\boldsymbol{x}_k)^{\mathrm{T}}(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) + \frac{1}{2}M\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|_2^2 \leq 0 \tag{2.128}$$

Given (2.125), proving (2.128) is equivalent to prove the following (as shown in Proof 2.5.4):

$$\|\nabla f^i(\boldsymbol{x}_k)\|_2 \cos\alpha - \|\bar{G}^i(\boldsymbol{x}_k, \nu_k; \xi^i)\|_2 \cos\beta \leq \sqrt{d} M \nu_k \tag{2.129}$$

where α is the angle between gradient $\nabla f^i(\boldsymbol{x}_k)$ and search direction $\boldsymbol{p}_k$, β is the angle between $\bar{G}^i(\boldsymbol{x}_k, \nu_k; \xi^i)$ and $\boldsymbol{p}_k$. Let ψ be the angle between $\bar{G}^i(\boldsymbol{x}_k, \nu_k; \xi^i) - \nabla f^i(\boldsymbol{x}_k)$ and $-\boldsymbol{p}_k$. Similar to Proof 2.5.4, there is

$$\left(\|\nabla f^i(\boldsymbol{x}_k)\|_2 \cos\alpha - \|\bar{G}^i(\boldsymbol{x}_k, \nu_k; \xi^i)\|_2 \cos\beta\right)^2 = \left(\|\Delta^i_{k^i,\xi^i}\|_2 \cos\psi\right)^2 \tag{2.130}$$

$$\|\nabla f^i(\boldsymbol{x}_k)\|_2 \cos\alpha - \|\bar{G}^i(\boldsymbol{x}_k, \nu_k; \xi^i)\|_2 \cos\beta \leq \|\Delta^i_{k^i,\xi^i}\|_2 \tag{2.131}$$

Since with $n_k = -\frac{16\sigma^2 \ln\delta}{3\nu_k^4 M^2}$, (2.112) holds, then

$$\mathbb{P}\left\{\|\nabla f^i(\boldsymbol{x}_k)\|_2 \cos\alpha - \|\bar{G}^i(\boldsymbol{x}_k, \nu_k; \xi^i)\|_2 \cos\beta \leq \sqrt{d} M \nu_k\right\} \geq 1 - \delta \tag{2.132}$$

thus

$$\mathbb{P}\left\{\bar{f}^i(\boldsymbol{x}_k; \xi_0^i) + \hat{G}^i(\boldsymbol{x}_k, \nu_k; \xi^i)^{\mathrm{T}}(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) + \frac{1}{2} M \|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|_2^2 \leq 0\right\} \geq 1 - \delta \tag{2.133}$$

□

Then, the safe set for $f^i(\boldsymbol{x}_k)$ at $\boldsymbol{x}_k$ with noisy measurements is modified as:

$$\hat{\mathcal{S}}^i_{k,\xi^i} := \left\{\boldsymbol{x} \in \mathbb{R}^d : \bar{f}^i(\boldsymbol{x}_k) + \hat{G}^i(\boldsymbol{x}_k, \nu_k; \xi^i)^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{x}_k) + \frac{1}{2} M \|\boldsymbol{x} - \boldsymbol{x}_k\|_2^2 \leq 0\right\} \tag{2.134}$$

This safe set is a ball centered at $\hat{\mathcal{O}}^i_{k,\xi^i}$ with radius $\hat{\mathcal{R}}^i_{k,\xi^i}$:

$$\hat{\mathcal{O}}^i_{k,\xi^i} = \boldsymbol{x}_k - \frac{1}{M} \hat{G}(\boldsymbol{x}_k, \nu_k; \xi^i) \tag{2.135}$$

$$\hat{\mathcal{R}}^i_{k,\xi^i} = \sqrt{\frac{1}{M}\left(\frac{\|\hat{G}^i(\boldsymbol{x}_k, \nu_k; \xi^i)\|_2^2}{M} - 2\bar{f}^i(\boldsymbol{x}_k; \xi_0^i)\right)} \tag{2.136}$$

The overall safe set $\mathcal{S}_{k,\xi}$ for all constraints $f^i(\boldsymbol{x})$, $i = 1, \ldots, m$ is the intersection of $\hat{\mathcal{S}}^i_{k,\xi^i}$

$$\mathcal{S}_{k,\xi} := \left\{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x} - \hat{\mathcal{O}}^i_{k,\xi^i}\|_2 \leq \hat{\mathcal{R}}^i_{k,\xi^i}, \forall = 1, \ldots, m\right\} \tag{2.137}$$

The search direction projection has the same procedure in section sec:direction projection for the case of exact measurements, except that the $\boldsymbol{C}_k$ matrix consists

of $\hat{G}^i(\boldsymbol{x}_k, \nu_k; \xi^0), i \in \mathcal{A}$ and $\boldsymbol{p}_k$ is $\bar{G}^0(\boldsymbol{x}_k, \nu_k; \xi^0)$.

In addition to the above modification, the step length selection method shall be adapted accordingly. The uncertain gradient estimators and function measurements are replaced by sample-average counterparts.

The step length upper bound in the Algorithm 4 is changed to the following:

$$\hat{\alpha}_{k,\xi} = \max\{\alpha \in \mathbb{R} : \|\boldsymbol{x}_k + \alpha \boldsymbol{p}_k - \hat{\mathcal{O}}^i_{k,\xi^i}\|_2 < \hat{\mathcal{R}}^i_{k,\xi^i}, \forall i = 1, \ldots, m, \alpha > 0\} \quad (2.138)$$

and the modified step length selection algorithm for noisy measurements is shown below:

Algorithm 6 Safe step length selection for noisy measurements (n-StepLength)

Input: Search direction $\boldsymbol{p}_k$, gradient estimator for objective $\bar{G}^0(\boldsymbol{x}_k, \nu_k; \xi^0)$, gradient estimator for constraints $\hat{G}^i(\boldsymbol{x}_k, \nu_k; \xi^i)$, centers and radii $\hat{\mathcal{O}}^i_{k,\xi^i}$, $\hat{\mathcal{R}}^i_{k,\xi^i}$, $i = 1, \ldots, m$, $\hat{\alpha}_{k,\xi} > 0$, $h > 0$, $n_k > 0$, $\rho \in (0, 1)$, $c \in (0, 1)$;
while $\min\limits_{i=1,\ldots,m} \left\{\hat{\mathcal{R}}^i_{k,\xi^i} - \| (\boldsymbol{x}_k + \hat{\alpha}_{k,\xi} \boldsymbol{p}_k) \|_2 - \hat{\mathcal{O}}^i_{k,\xi^i}\right\} \leq 0$ **do**
    $\hat{\alpha}_{k,\xi} = \rho \hat{\alpha}_{k,\xi}$;
end while
Set $\alpha_{k,\xi} \leftarrow \hat{\alpha}_{k,\xi}$;
Compute $\bar{f}^i(\boldsymbol{x}_k; \xi^i_0)$ and $\bar{f}^i(\boldsymbol{x}_k + \alpha_{k,\xi} \boldsymbol{p}_k; \xi^i_0),\ i = 0, \ldots, m$ using (2.107)
while $\bar{f}^0(\boldsymbol{x}_k + \alpha_{k,\xi} \boldsymbol{p}_k; \xi^0_0) \geq \bar{f}^0(\boldsymbol{x}_k; \xi^0_0) + c \alpha_{k,\xi} \bar{G}^0(\boldsymbol{x}_k, \nu_k; \xi^0)^{\mathrm{T}} \boldsymbol{p}_k$ **or** $\min\limits_{i=1,\ldots,m} \left\{-\bar{f}^i(\boldsymbol{x}_k + \alpha_{k,\xi} \boldsymbol{p}_k; \xi^i_0))\right\} < h$ **do**
    $\alpha_{k,\xi} = \rho \alpha_{k,\xi}$;
    Update $\bar{f}^i(\boldsymbol{x}_k + \alpha_{k,\xi} \boldsymbol{p}_k; \xi^i_0),\ i = 0, \ldots, m$;
end while
Return: $\alpha_{k,\xi}$;

2.6.3 Safe-line-search algorithm (n-SLS) formulation

With the above modifications, the safe-line-search algorithm for noisy measurements is formulated below.

Algorithm 7 Safe-line-search algorithm for noisy measurements (n-SLS)

Input: $\boldsymbol{x}_0 \in \mathcal{D}$, $f^i(\boldsymbol{x}_0), i = 0, \dots, m$, L, $M > 0$, $d > 0$, $\mu > 0$, $h > 0$, $\epsilon > 0$, $\sigma > 0$, $\rho \in (0, 1)$, $c \in (0, 1)$, $\delta \in [0, 1]$;
for $k = 0, 1, \dots$ **do**
  Take $\nu_k = \min\left\{\frac{2\mu}{\sqrt{d}M}, \frac{\min\limits_{i=1,\dots,m}\{-\bar{f}^i(\boldsymbol{x}_k;\xi_0^i)\}}{2L}\right\}$, $n_k = -\frac{16\sigma^2 \ln\delta}{3\nu_k^4 M^2}$;
  For each $f^i(\boldsymbol{x})$, $i = 0, \dots, m$, make evaluations at points $\boldsymbol{x}_k + \nu_k e_j$, $j = 1, \dots, d$, measure each value by n_k times;
  Compute the gradient estimators $\bar{G}^i(\boldsymbol{x}_k, \nu_k; \xi^i)$ of $\nabla f^i(\boldsymbol{x}_k)$, $i = 0, \dots, m$ using (2.109);
  Set $\boldsymbol{p}_{k,\xi} = -\bar{G}^0(\boldsymbol{x}_k, \nu_k; \xi^0)$ **or** *Quasi-Newton* direction (2.42);
  Compute $\hat{G}^i(\boldsymbol{x}_k, \nu_k; \xi^i)$, $i = 1, \dots, m$ (2.126);
  Compute the centers $\hat{\mathcal{O}}^i_{k,\xi^i}$ and radii $\hat{\mathcal{R}}^i_{k,\xi^i}$ of safe set $\hat{\mathcal{S}}^i_{k,\xi^i}$, $i = 1, \dots, m$ (2.135), (2.136);
  if $\min\{-\bar{f}^i(\boldsymbol{x}_k; \xi_0^i)\} \le h$ **then**
    $\mathcal{A} := \left\{i : -\bar{f}^i(\boldsymbol{x}_k; \xi_0^i)\} \le h\right\}$;
    $\boldsymbol{C}_k = \left[\hat{G}^i(\boldsymbol{x}_k, \nu_k; \xi^i), \dots, \hat{G}^j(\boldsymbol{x}_k, \nu_k; \xi^i)\right]$, $i, \dots, j \in \mathcal{A}$;
    $\boldsymbol{\lambda}_{sol} \leftarrow$ Solve Non-negative LS problem 3;
    $\boldsymbol{p}^{proj}_{k,\xi} = \boldsymbol{p}_{k,\xi} - \boldsymbol{C}_k \boldsymbol{\lambda}_{sol}$;
    $\boldsymbol{p}_{k,\xi} = \boldsymbol{p}^{proj}_{k,\xi}$;
  end if
  Find $\alpha_{k,\xi} \leftarrow$ n-StepLength 6;
  Set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_{k,\xi}\boldsymbol{p}_{k,\xi}$;
  Evaluate $f^i(\boldsymbol{x}_{k+1})$, $i = 0, \dots, m$, measure each value by n_k times;
  if $\boldsymbol{x}_{k+1} - \boldsymbol{x}_k \le \epsilon$ **then**
    $\boldsymbol{x}_* = \boldsymbol{x}_k$;
    $\boldsymbol{\lambda}_* = \boldsymbol{0}$;
    if $\mathcal{A} \neq \emptyset$ **then**
      $\lambda^i_* = \lambda^i_{sol}$, $i \in \mathcal{A}$;
    end if
    Algorithm Stop;
  end if
end for

2.7 Convergence and safety analysis

2.7.1 Convergence analysis

Theorem 7. *With sufficiently large iterations, the e-SLS converges to a KKT point.*

Proof. *If the optimal point $\boldsymbol{x}_*$ is inside of the feasible set $\mathcal{D}$, recall that iteration has the form*

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k \tag{2.139}$$

where α_k is selected by Algorithm 4, and $\boldsymbol{p}_k$ is a descend direction.

Let ψ_k be the angle between $\boldsymbol{p}_k$ and negative gradient $-G^0(\boldsymbol{x}_k, \nu_k)$, then as k increases, there is

$$\sum_{k>0} \cos^2 \psi_k \|G^0(\boldsymbol{x}_k, \nu_k)\|^2 < \infty \tag{2.140}$$

If $\boldsymbol{p}_k$ is the steepest descend direction, then $\cos \psi_k = 1$, $k > 0$, therefore

$$\sum_{k>0} \|G^0(\boldsymbol{x}_k, \nu_k)\|^2 < \infty \tag{2.141}$$

which implies that

$$\lim_{k\to\infty} \|G^0(\boldsymbol{x}_k, \nu_k)\| = 0 \tag{2.142}$$

Thus, the point $\boldsymbol{x}_k = \arg\left(\lim_{k\to\infty} \|G^i(\boldsymbol{x}_k, \nu_k)\|\right)$, and the Lagrangian multipliers $\boldsymbol{\lambda}_k = \mathbf{0}$, is a KKT point $[\boldsymbol{x}_k, \mathbf{0}]$.

If the optimal point is outside of feasible set $\mathcal{D}$, let $\mathcal{A}$ be the set of indices of approximately active constraints at the optimal feasible point $\boldsymbol{x}_*$, that is:

$$\mathcal{A} := \left\{ i : \ -f^i(\boldsymbol{x}_*) \le h \right\} \tag{2.143}$$

where h is a safety threshold. From line 9 of Algorithm 5, the descent direction $\boldsymbol{p}_k$ is projected on $\boldsymbol{p}_k^{proj}$. From Proof 2.5.5, it is known that

$$\measuredangle(\boldsymbol{p}_k^{proj}, \boldsymbol{p}_k) \le \frac{\pi}{2} \tag{2.144}$$

and

$$\frac{\pi}{2} \le \measuredangle(\boldsymbol{p}_k^{proj}, G^i(\boldsymbol{x}_k, \nu_k)) \le \frac{3\pi}{2}, \ \forall i \in \mathcal{A} \tag{2.145}$$

That is, the direction $\boldsymbol{p}_k^{proj}$ is a descent direction without increasing active constraints. From line 18 of the algorithm, if the convergence condition is satisfied at point $\boldsymbol{x}_k$, then $\alpha_k \boldsymbol{p}_k^{proj} \to 0$, thus either $\alpha \to 0$ or $\|\boldsymbol{p}_k^{proj}\|_2 \to 0$.

If $\alpha \to 0$, then from the Armijo condition (2.95), there is:

$$f^0(\boldsymbol{x}_k + 0_+ \boldsymbol{p}_k^{proj}) \ge f^0(\boldsymbol{x}_k) + 0_+ c \nabla f^0(\boldsymbol{x}_k)^{\mathrm{T}} \boldsymbol{p}_k^{proj} \tag{2.146}$$

$$f^0(\boldsymbol{x}_{k-1}) \geq f^0(\boldsymbol{x}_k) \tag{2.147}$$

that is, all points around $\boldsymbol{x}_k + 0_+\boldsymbol{p}_k^{proj}$ *have a larger value than* $f^0(\boldsymbol{x}_k)$*, and*

$$\nabla f^0(\boldsymbol{x}_{k-1})^{\mathrm{T}}\boldsymbol{p}_{k-1}^{proj} \leq 0 \tag{2.148}$$

$$\nabla f^0(\boldsymbol{x}_k + 0_+\boldsymbol{p}_k^{proj})^{\mathrm{T}}\boldsymbol{p}_k^{proj} \geq 0 \tag{2.149}$$

which implies that $\|\boldsymbol{p}_k^{proj}\|_2 \to 0$*.*

Thus, from $\alpha_k\boldsymbol{p}_k^{proj} \to 0$*, there is* $\|\boldsymbol{p}_k^{proj}\|_2 \to 0$*. Then, from line 13 of the algorithm, there is* $\boldsymbol{p}_k \approx \boldsymbol{C}_k\boldsymbol{\lambda}_{sol}$*, which means that the gradient* $-G^0(\boldsymbol{x}, \nu_k)$ *can be linearly expressed by* $G^i(\boldsymbol{x}_k, \nu_k)$*,* $i \in \mathcal{A}$*. From Lemma 4 we know that* $[\boldsymbol{x}_k, \boldsymbol{\lambda}_{sol}, \boldsymbol{0}]$ *is a KKT point.* □

2.7.2 Safety analysis

Theorem 8. *With gradient estimator* $\hat{G}^i(\boldsymbol{x}_k, \nu_k) = G^i(\boldsymbol{x}_k, \nu_k) + \frac{\sqrt{d}M\nu_k\|\boldsymbol{p}_k\|_2}{2\boldsymbol{e}^{\mathrm{T}}\boldsymbol{p}_k}\boldsymbol{e}$*, and safety threshold* $h \geq 0$*, e-SLS is safety-guarantee during optimization.*

Proof. *Assume that the unconstrained optimal point* $\boldsymbol{x}_*^{unc}$ *of the objective function* $f^0(\boldsymbol{x})$ *is outside of the feasible set* $\mathcal{D}$*,* $f^i(\boldsymbol{x})$*,* $i \in \mathcal{A}$ *are the active constraints at the optimal feasible solution* $\boldsymbol{x}_*$*. By Algorithm 4, there is:*

$$\|\boldsymbol{x}_k + \alpha_k\boldsymbol{p}_k - \hat{\mathcal{O}}_k^i\|_2 < \hat{\mathcal{R}}_k^i, \forall i = 1, \ldots, m \tag{2.150}$$

where $\hat{\mathcal{O}}_k^i$ *and* $\hat{\mathcal{R}}_k^i$ *are the center and radius of safe sets* $\hat{\mathcal{S}}_k^i$ *computed with* $\hat{G}^i(\boldsymbol{x}_k, \nu_k) = G^i(\boldsymbol{x}_k, \nu_k) + \frac{\sqrt{d}M\nu_k\|\boldsymbol{p}_k\|_2}{2\boldsymbol{e}^{\mathrm{T}}\boldsymbol{p}_k}\boldsymbol{e}$*.*

If $\boldsymbol{x}_k$ *is not close to a boundary, that is* $-f^i(\boldsymbol{x}_k) > h$*,* $\forall i = 1, \ldots, m$*, from Theorem 3, there is:*

$$f^i(\boldsymbol{x}_k + \alpha_k\boldsymbol{p}_k) \leq f^i(\boldsymbol{x}_k) + \alpha_k\nabla f^i(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{p}_k + \frac{1}{2}M\alpha_k^2\|\boldsymbol{p}_k^2\| \leq 0 \tag{2.151}$$

that is, when $\boldsymbol{x}_k$ *is not close to a boundary, all* $\boldsymbol{x}_k$ *are safe.*

If there are some $-f^i(\boldsymbol{x}_k) \leq h$*, then from line 5 in Algorithm 4, the selected step length* α_k *will be such that:*

$$f^i(\boldsymbol{x} + \alpha_k\boldsymbol{p}_k) \leq -h < 0 \tag{2.152}$$

Thus, it can be guaranteed that $\max\{f^i(\boldsymbol{x})\} \leq -h < 0$*,* $\forall k$*. that is, e-SLS is safe during the optimization process.* □

2.8 Simulation analysis with case studies

2.8.1 Problem description

In this section, the safe line search algorithm for exact measurements and noisy measurements are tested on simulations, and a comparison to other algorithms including the 0-LBM method [UKK19], the DFO-TR method and the SafeOpt-MC method [BKS21] is conducted.

In the simulation, both objective and constraint functions are unknown to the algorithms, while they can only access the function measurements. In addition, the safety requirement that during the optimization process, no point that violates the safety constraints is evaluated shall be fulfilled. Hence it is a model-free safe optimization problem.

The simulation aims to minimize a $2d$ function with different constraints. There are 2 cases in the simulation. In the first case, the constraints are box constraints on the decision variables. In the second case the constraint is changed to a general nonlinear constraint. Both cases are test with exact measurements and noisy measurement respectively.

As this simulation test mainly focus on comparison of the convergence rate of different algorithms, thus a convex objective function is used so that only one global optimum exists. However, the objective function can be a non-convex function. In this case, the algorithm only finds a local minimum and the found minimum point depends on the starting point.

2.8.2 Case 1 Optimization problem with box constraints

In the first case, the algorithm is test on the following optimization problem:

$$\min_{x\in\mathbb{R}^2} \quad (x_1 - 2.7)^2 + 0.5(x_2 - 0.5)^2 - 5 \tag{2.153}$$

$$\text{subject to} \quad x_1 \leq 2.7 \tag{2.154}$$

$$x_2 \geq -5 \tag{2.155}$$

The objective is a convex function with optimal point located at $[2.7, 0.5]$ with optimal value -5. The constraints are box constraints on decision variables.

Firstly, the e-SLS algorithm is tested for solving this problem. The optimization trajectory of e-SLS method is shown in Figure 2.6:

In Figure 2.6, the blue and cyan circles represent the safe set at each iteration for constraints $x_1 \leq 2.7$ and $x_2 \geq -5$. The safe set for one constraint dose not necessarily guarantee safety of other constraints, however, the decision point at each iteration is chosen in the intersection of all safe sets for every constraints, thus the decision points during the optimization process are safe. Besides, one

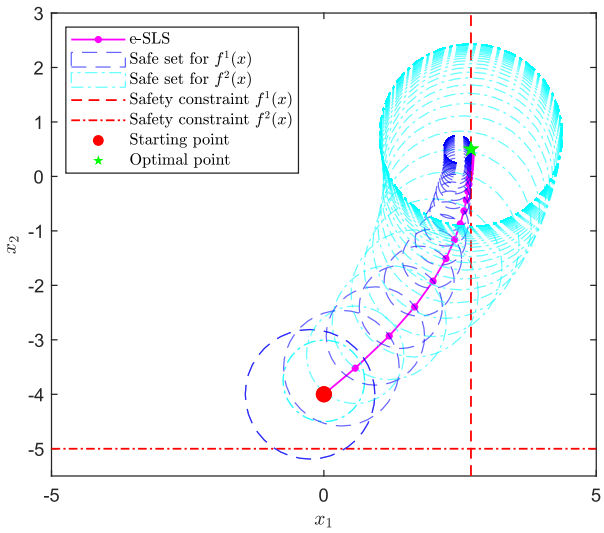


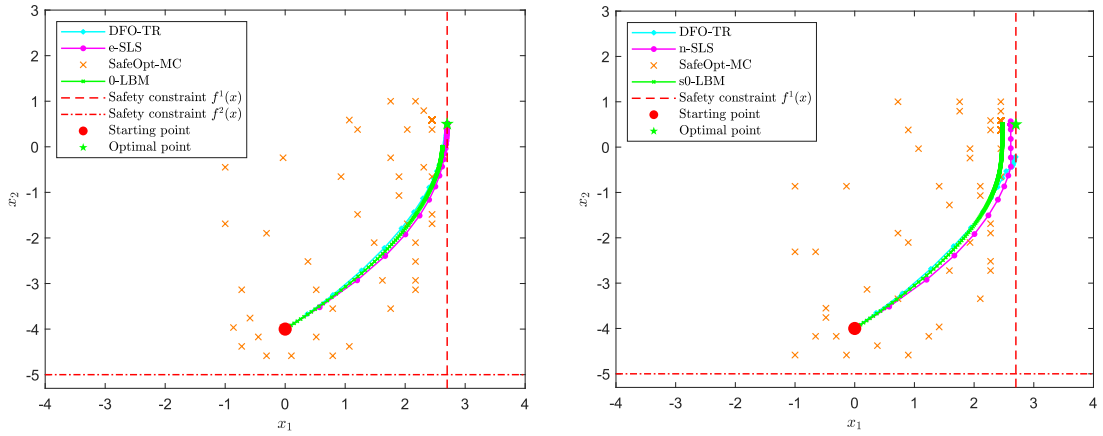
Figure 2.6: Optimization trajectory of e-SLS method in case 1. The red point denotes the starting point, the green star denotes the optimal point, the magenta line is the optimization trajectory of e-SLS method. The red dash lines are the safety constraints, blue and cyan dash circles denote the safe sets at each iteration for safety constraints $f^1(\boldsymbol{x}) \leq 0$ and $f^2(\boldsymbol{x}) \leq 0$ respectively.

can see that the safe set shrinks as the decision point approaching to the corresponding boundary since the radius of safe set depends on the constraint value $f^i(\boldsymbol{x})$. As it converges to 0, the radius reduces. Because every decision points are chosen inside the common parts of all safe sets, throughout the optimization process, the safety constraints are not violated.

Next, a comparison of different algorithms is taken. This comparison includes the DFO-TR method, the e-SLS method, the SafeOpt-MC method and the 0-LBM method. Figure 2.7 shows the optimization trajectories of different algorithms. Figure 2.8 shows the convergence of objective function in different algorithms and Figure 2.9 illustrates the constraint function values.

In Figure 2.7a one can see that the DFO-TR method, e-SLS method perform similar optimization trajectories while the 0-LBM method has a similar trajectory trend to the preceding two methods but with a farther distance to the optimal point when it converges. The SafeOpt-MC method samples a set of safe points to explore the underlying objective and constraint functions. And it repeats sampling the points that give the minimum values in the objective function, which is the points near to the optimal point, after learning the underlying functions. It can be noticed that all methods take measurements within the underlying safe region.

From Figure 2.7b it can seen that, the n-SLS method and s0-LBM method perform a more conservative optimization trajectories compared to the results with exact measurements. They keep a larger distance to the boundary to re-



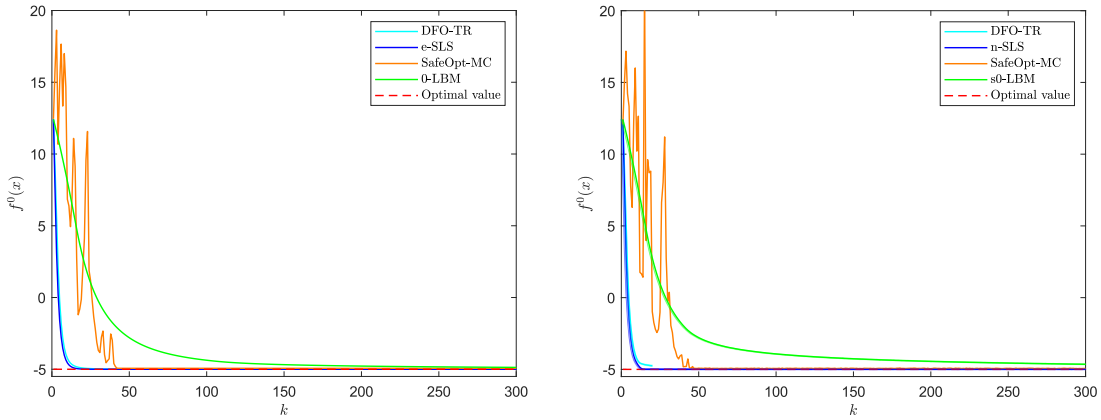
(a) Optimization trajectories of case 1 with exact measurements

(b) Optimization trajectories of case 1 with noisy measurements

Figure 2.7: (a) Optimization trajectory of different algorithms with exact measurements. The red point denotes the starting point, green star denotes the optimal point, red dash lines are the safety constraints. Cyan, magenta, and green lines are the optimization trajectories of DFO-TR method, e-SLS method and 0-LBM method respectively. The orange crosses denote the sample points of SafeOpt-MC method. (b) Optimization trajectories of different algorithms with measurement noise $\xi \sim \mathcal{N}(0, 0.01^2)$.

duce the probability that it violates the safety constraints. The SafeOpt-MC has an intrinsic advantage of handling measurement noises, because it uses Gaussian process to explore and learn the underlying functions, and it considers the influence of measurement noises on the model learning by adding additive adjustment to the covariance matrix of the Gaussian process. Thus, it increases the variance of value of each measurement. The DFO-TR method is influenced by the measurement noises, and it stops updating when approaching to a boundary. This is because that the trust region is computed using estimated gradients computed by finite difference method, which is sensitive to measurement noises. With incorrect trust regions, the optimization problem may be infeasible to the problem solvers.

From Figure 2.8a it can be seen that the e-SLS and DFO-TR methods have similar convergence performance: they both converge to the optimal point within a small number of iterations, and the optimality gap to the true optimal value is small. These two methods have the fastest convergence rate among all methods. The 0-LBM method requires much larger number of iterations to converge to the optimal point, since the step length of 0-LBM method exponentially decreases as it approaches to a boundary. Thus, the closer it is to a boundary, the slower it converges. The SafeOpt-MC method does not provide a monotonous decrease in the objective function, since it first explores the decision space by choosing the



(a) Convergence of objective function of case 1 with exact measurements

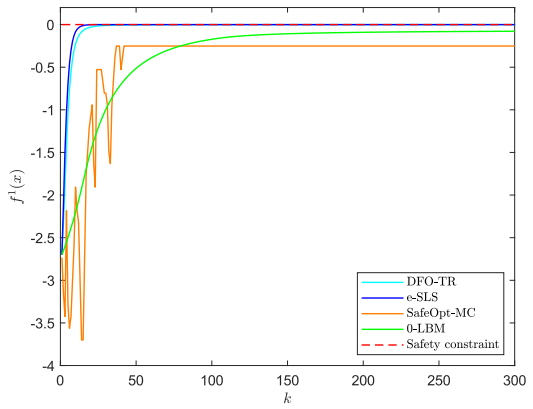
(b) Convergence of objective function of case 1 with noisy measurements

Figure 2.8: (a) Convergence of objective function with exact measurements. Cyan, blue, orange and green lines are the convergence result of DFO-TR method, e-SLS method, SafeOpt-MC method and 0-LBM method respectively. The red dash line is the ground true optimal value. (b) Convergence of objective function with measurement noise $\xi \sim \mathcal{N}(0, 0.01^2)$.

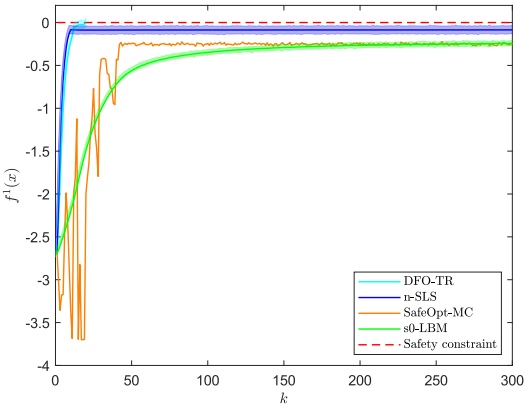
points with the highest uncertainty, then it exploits the function measurements by choosing points that minimize the objective function. During the exploration phase, the algorithm possibly samples points that give higher values in the objective function. Besides, the SafeOpt-MC method discretizes the decision space, therefore, the optimization performance and the computation time much depend on the dimensions of decision space and the density of discretization in each dimension. In this simulation, the search range of SafeOpt-MC method is $[-1, 3]$ in x_1 and $[-5, 1]$ in x_2, and the discretization is taken with 30 points in each dimension. Figure 2.8b shows that the n-SLS, SafeOpt-MC, the s0-LBM methods still perform a good convergence result in the presence of measurement noises, while the DFO-TR method stop converging at early iteration as it contact a boundary. Thus the convergence becomes degraded given the measurement noises.

From Figure 2.9a one can see that all methods are safety guaranteed. The DFO-TR and e-SLS methods approach to the boundary faster than other two methods and in convergence, they are both closer to the boundary as well. The SafeOpt-MC method has the largest distance to the boundary and the 0-LBM method converges lastly. Figure 2.9b shows that the DFO-TR method has some measurements slightly beyond the constraint, while all other methods keep within the safe region.

In addition to the measure of number of iterations, the computation time is another key factor that evaluates an algorithm. As the number of iterations can not present the computation burden of algorithms. Table 2.8.2 shows the



(a) Constraint function of case 1 with exact measurements



(b) Constraint function of case 1 with noisy measurements

Figure 2.9: (a) Constraint function values with exact measurements. Cyan, blue, orange and green lines are the convergence result of DFO-TR method, e-SLS method, SafeOpt-MC method and 0-LBM method respectively. The red dash line is the constraint upper bound. (b) Constraint function values with measurement noise $\xi \sim \mathcal{N}(0, 0.01^2)$.

number of iterations, computation time to convergence and the optimiality gap to the true optimal value of different algorithms with exact measurements.

Table 2.1: Optimization performance of different algorithms

Algorithms	Number of iterations to convergence	Computation time to convergence	Optimality Gap
DFO-TR	22	28.071s	0.240%
e-SLS	**19**	0.060s	**0.083%**
SafeOpt-MC	42	35.872	1.342%
0-LBM	300	**0.031s**	2.601%

From Table 2.8.2 one can see that the SafeOpt-MC method has the least number of iterations, while the 0-LBM has the largest. The e-SLS has a comparable result to the SafeOpt-MC method. For the computation time, however, the SafeOpt-MC requires the longest computation time, since updating the Gaussian process model is time consuming, and it is exponentially growth with the number of dimensions of the problem. The 0-LBM has the fastest computation time. For the optimality gap, the e-SLS method has the lowest deviation to the true optimal value among all methods. From this table, it can be concluded that the e-SLS method performs a good trade-off between the optimization performance and the computation time.

Lastly, to study the effect of starting point near to a boundary on the conver-

gence performance of different algorithms, one complementary simulation is run with an starting point at $\boldsymbol{x}_{start} = [0, -4.99]$, which is further close to a boundary. The simulation results is shown in Figure 2.10.

By this example, one can see that the convergence performances of DFO-TR, e-SLS and SafeOpt-MC methods are not much affected by the changing of starting point to the vicinity of a boundary, while the convergence of 0-LBM method suffer a large lag in the beginning of the optimization. The reason for the lag is that the step length of 0-LBM method is small when it is close to a boundary, regardless the search direction toward deviating the boundary. The DFO-TR and e-SLS methods build a safe region around current point, and safe region has larger space on the side away from the boundary than the side towards the boundary. Thus, if the search direction deviates from the boundary, then the algorithm is not restricted by the small distance to the boundary and still produces a larger step along the search direction. This is one of advantages of DFO-TR and e-SLS methods over the 0-LBM method.

In the next section, the study case 2 with nonlinear constraints is analysed.

2.8.3 Case 2 Optimization problem with nonlinear constraints

In the second case, the algorithms are tested on the following optimization problem:

$$\min_{\boldsymbol{x}\in\mathbb{R}^2} \quad (x_1 - 2.7)^2 + 0.5(x_2 - 0.5)^2 - 5 \tag{2.156}$$

$$\text{subject to} \quad 1.5\sin(x_1) - x_2 \leq 0 \tag{2.157}$$

In this problem, the objective function remains the same as in the case 1, while the constraint is changed to a general nonlinear constraint. The optimization result is shown in the following figures. Figure 2.11 shows the optimization trajectories of all algorithms, Figure 2.12 shows the convergence of objective function and Figure 2.13 illustrates the constraint values of all algorithms in the optimization.

For optimization problem with non-linear constraint, one question is how to approach to the optimal point if it is obstructed by non-linear constraint?

The 0-LBM method utilizes the barrier term in the barrier function to perform a sliding optimization trajectory along the boundary towards the optimal point. Because, when decision points are close to a boundary, the barrier term considerably increases the value of barrier function, thus minimizing the barrier function equivalently repels the decision points away from the boundary. By searching for points producing a less value in the barrier function, the optimization trajectory goes along the boundary and approaches to the optimal point. However, the optimization performance of 0-LBM method depends on the choice of barrier coefficient. Ideally, the coefficient shall gradually decreases to zero,

while if the decreasing speed is too large, it approaches to a boundary too early before it is close to the optimal point. On the other hand, if the decreasing speed is too slow, the convergence rate can be slow because the large weighting factor on the barrier term keeps the decision points away from the boundary, which the optimal point is close to.

The e-SLS method uses gradient projection to tackle with non-linear constraint. When the optimization trajectory is close to a boundary below a threshold h, the algorithm projects the search direction on the tangent of the boundary, so that at that iteration, the algorithm searches in the direction parallel to the boundary and towards the optimal point. In this case, the optimization performance depends on the value of h, the less h is, the less optimality gap it is in convergence.

The SafeOpt-MC method is not affected by the shape of constraint function, because it is a global optimization method that it firstly explores possibly the entire decision space before making samples that minimize the objective function.

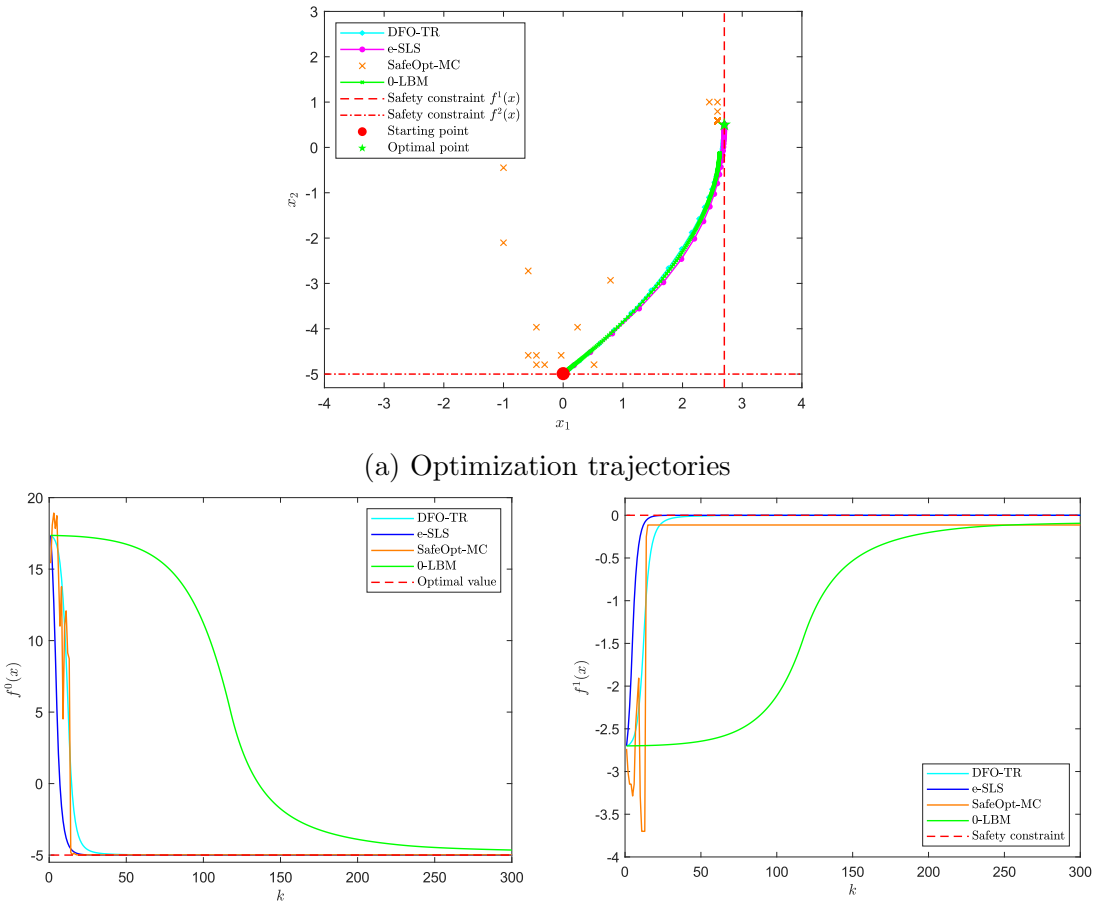
The DFO-TR method is not suitable for non-linear constrained optimization problem, since it minimizes solely the objective function while not considers penalizing being too close to the boundary. Thus, the optimization trajectory can be obstructed by non-linear constraints.

From Figure 2.11 one can see that both e-SLS and 0-LBM method produce sliding optimization trajectories along the non-linear boundary approaching to the optimal point. The SafeOpt-MC method finds the optimal point after some iterations of exploration in the decision space. The DFO-TR method directly goes in the direction towards to the optimal point and it is obstructed by a non-linear boundary between the starting point and the optimal point.

From Figure 2.12, it can be seen that though the e-SLS method and 0-LBM method has similar optimization trajectories, the number of iteration in 0-LBM method is much larger than the values of e-SLS method, as the same in the first simulation case. Both e-SLS and SafeOpt-MC methods converge to the optimal point within 50 iterations.

By Figure 2.13 one can see that the e-SLS, SafeOpt-MC and 0-LBM method are safe during optimization. The DFO-TR method is safe with exact measurements while some measurements of it in the presence of measurement noises are unsafe.

In the next chapter, the study of optimal power flow problem is introduced. This problem generally is a large-scale non-linear, non-convex optimization problem. For system with smaller scale, the problem can be solved with model-based optimization approaches, however, for some power systems, the exact model of it is either not available or is not accurate. In this case, model-free optimization techniques are used for solving the problem. After introducing the optimal power flow problem, the e-SLS method is implemented for solving the problem and its performance is analysed.

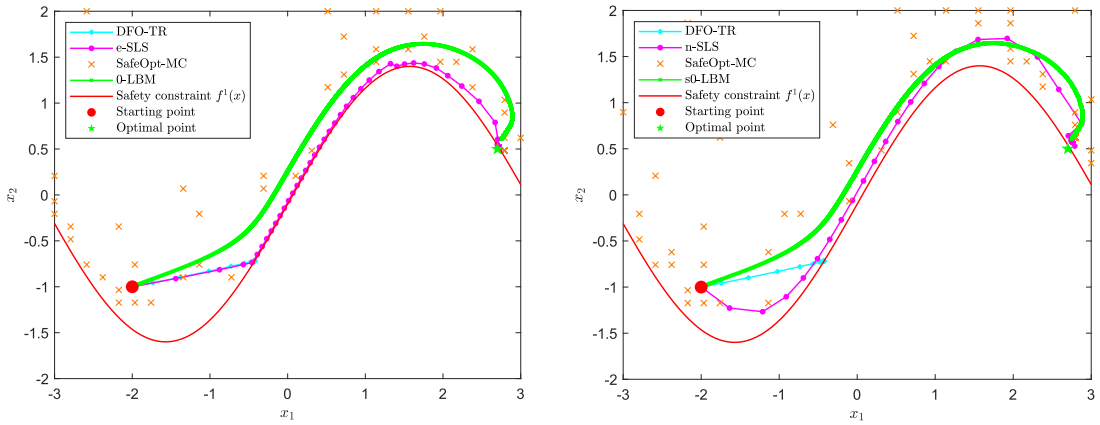


(a) Optimization trajectories

(b) Convergence of objective function

(c) Constraint function values

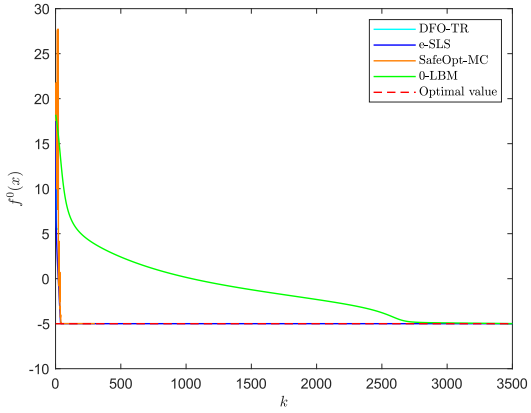
Figure 2.10: Optimization result of different algorithms with exact measurements and starting point at $\boldsymbol{x}_{start} = [0, -4.99]$. Cyan, magenta, green lines represent the result of DFO-TR method, e-SLS method and 0-LBM method respectively. The orange crosses are the sample points of SafeOpt-MC method. (a) shows the optimization trajectories of different algorithms, (b) is the convergence of objective and (c) is the constraint function values.



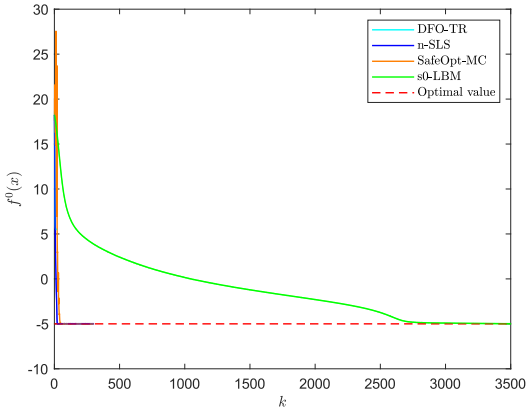
(a) Optimization trajectories of case 2 with exact measurements

(b) Optimization trajectories of case 2 with noisy measurements

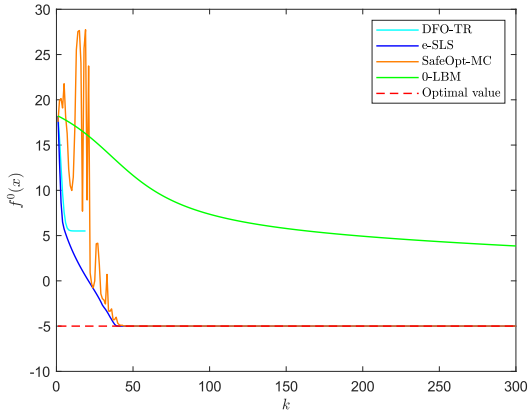
Figure 2.11: (a) Optimization trajectory of different algorithms with exact measurements. The red point denotes the starting point, green star denotes the optimal point, red curve is the non-linear safety constraints. Cyan, magenta, and green lines are the optimization trajectories of DFO-TR method, e-SLS method and 0-LBM method respectively. The orange crosses denote the sample points of SafeOpt-MC method. (b) Optimization trajectories of different algorithms with measurement noise $\xi \sim \mathcal{N}(0, 0.01^2)$.



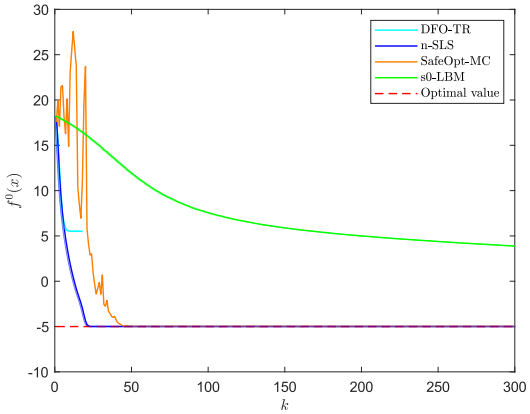
(a) Convergence of objective function of case 2 with exact measurements



(b) Convergence of objective function of case 2 with noisy measurements

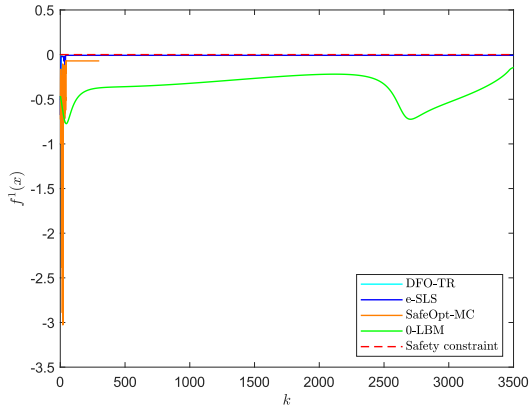


(c) The first 300 iterations of (a)

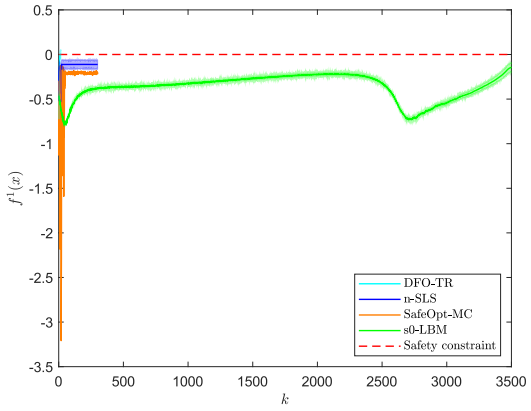


(d) The first 300 iterations of (b)

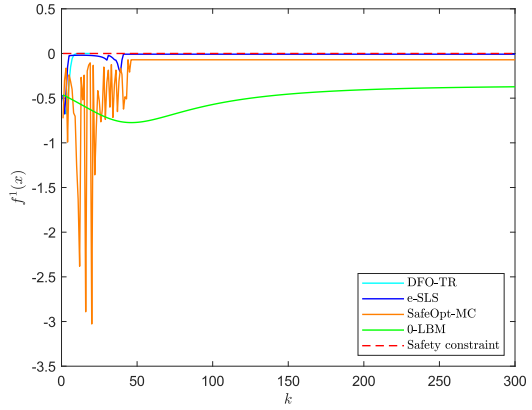
Figure 2.12: (a) Convergence of objective function with exact measurements. Cyan, blue, orange and green lines are the convergence result of DFO-TR method, e-SLS method, SafeOpt-MC method and 0-LBM method respectively. The red dash line is the ground true optimal value. (b) Convergence of objective function with measurement noise $\xi \sim \mathcal{N}(0, 0.01^2)$. (c) and (d) The first 300 iterations of optimization with exact measurements and noisy measurements respectively.



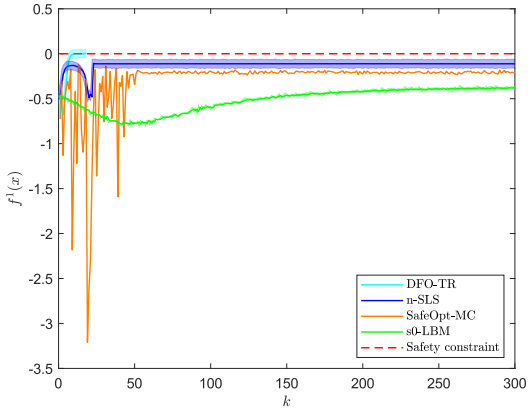
(a) Constraint function values of case 2 with exact measurements



(b) Constraint function values of case 2 with noisy measurements



(c) The first 300 iterations of (a)



(d) The first 300 iterations of (a)

Figure 2.13: (a) Constraint function values with exact measurements. Cyan, blue, orange and green lines are the convergence result of DFO-TR method, e-SLS method, SafeOpt-MC method and 0-LBM method respectively. The red dash line is the constraint upper bound. (b) Constraint function values with measurement noise $\xi \sim \mathcal{N}(0, 0.01^2)$. (c) and (d) The first 300 iterations of optimization with exact measurements and noisy measurements respectively.