Newton-Type Constrained Optimization
The Generalized Gauss-Newton Method

Mario Zanon
1. Nonlinear Programming and SQP

1. Nonlinear Programming and SQP

Newton’s Method

Problem: Find the zeros of $F(w)$

Newton’s Method: Linearize and iteratively solve

$$F(w_k) + \nabla F(w_k)^T p_k = 0$$
**Newton’s Method**

**Problem:** Find the zeros of $F(w)$

**Newton’s Method:** Linearize and iteratively solve

$$F(w_k) + \nabla F(w_k)^T p_k = 0$$

---

**Unconstrained Optimization**

**Problem:** $\min_w f(w)$

First order necessary conditions (FONC): $\nabla f(w) = 0$

Find the zeros of FONC: Iteratively solve

$$\nabla f(w_k) + \nabla^2 f(w_k) p_k = 0$$
Nonlinear Programming Problem (NLP)

\[
\begin{align*}
\text{minimize} & \quad f(w) \\
\text{subject to} & \quad g(w) = 0 \\
& \quad h(w) \geq 0
\end{align*}
\]

(1)
Nonlinear Programming Problem (NLP)

\[
\begin{align*}
\text{minimize} & \quad f(w) \\
\text{subject to} & \quad g(w) = 0 \\
& \quad h(w) \geq 0
\end{align*}
\]

(1)

Newton Type Algorithm

Given an initial guess \( w_0 \), keep iterating:
Nonlinear Programming Problem (NLP)

\[
\begin{align*}
\text{minimize } & \quad f(w) \\
\text{subject to } & \quad g(w) = 0 \\
& \quad h(w) \geq 0
\end{align*}
\]  

(1)

Newton Type Algorithm

Given an initial guess \(w_0\), keep iterating:

1. determine a (descent) direction \(p_k\)
Nonlinear Programming and SQP

Nonlinear Programming Problem (NLP)

\[
\begin{align*}
\text{minimize} & \quad f(w) \\
\text{subject to} & \quad g(w) = 0 \\
& \quad h(w) \geq 0
\end{align*}
\]

(1)

Newton Type Algorithm

Given an initial guess \( w_0 \), keep iterating:

1. determine a (descent) direction \( p_k \)
2. determine a step length \( \alpha_k \)
Nonlinear Programming Problem (NLP)

\[
\begin{align*}
\text{minimize} & \quad f(w) \\
\text{subject to} & \quad g(w) = 0 \\
& \quad h(w) \geq 0
\end{align*}
\]  \tag{1}

Newton Type Algorithm

Given an initial guess \( w_0 \), keep iterating:

1. determine a (descent) direction \( p_k \)
2. determine a step length \( \alpha_k \)
3. compute the step: \( w_{k+1} = w_k + \alpha_k p_k \)
Nonlinear Programming Problem (NLP)

\[
\begin{align*}
\text{minimize} & \quad f(w) \\
\text{subject to} & \quad g(w) = 0 \\
& \quad h(w) \geq 0
\end{align*}
\] (1)

Newton Type Algorithm

Given an initial guess \( w_0 \), keep iterating:

1. determine a (descent) direction \( p_k \)
2. determine a step length \( \alpha_k \)
3. compute the step: \( w_{k+1} = w_k + \alpha_k p_k \)
4. check for convergence and return the solution
Nonlinear Programming Problem (NLP)

\[
\begin{align*}
\text{minimize} \quad & f(w) \\
\text{subject to} \quad & g(w) = 0 \\
& h(w) \geq 0
\end{align*}
\]

(1)

Lagrangian Function

\[
\mathcal{L}(w, \lambda, \mu) = f(w) - \lambda^T g(w) - \mu^T h(w)
\]
Nonlinear Programming Problem (NLP)

\[
\begin{align*}
\text{minimize} & \quad f(w) \\
\text{subject to} & \quad g(w) = 0 \\
& \quad h(w) \geq 0
\end{align*}
\]

(1)

Lagrangian Function

\[
\mathcal{L}(w, \lambda, \mu) = f(w) - \lambda^T g(w) - \mu^T h(w)
\]

1\textsuperscript{st} Order Necessary Conditions: the KKT system

\[
\begin{align*}
\nabla_w \mathcal{L}(w^*, \lambda^*, \mu^*) &= \nabla f(w^*) - \nabla g(w^*) \lambda^* - \nabla h(w^*) \mu^* = 0 \\
\nabla_\lambda \mathcal{L}(w^*, \lambda^*, \mu^*) &= g(w^*) = 0 \\
\nabla_\mu \mathcal{L}(w^*, \lambda^*, \mu^*) &= h(w^*) \geq 0 \\
\mu^* &\geq 0 \\
\mu^*^T h(w^*) &= 0
\end{align*}
\]
Without Inequalities

With Inequalities
Nonlinear Programming and SQP

**Without Inequalities**

- Linearize the KKT system

\[
\begin{bmatrix}
\nabla_w^2 \mathcal{L} & \nabla g \\
\n\nabla g^T & 0
\end{bmatrix}
\begin{bmatrix}
\Delta w_k \\
\lambda_{k+1}
\end{bmatrix}
= -
\begin{bmatrix}
\nabla f \\
\n\end{bmatrix}
\]

- Solve the linear system

---

**With Inequalities**

The last 3 KKT conditions are nonsmooth. At each iteration solve the QP

\[
\minimize \Delta w_k \quad \frac{1}{2} \Delta w_k^T \nabla_w^2 \mathcal{L} \Delta w_k + \nabla f^T \Delta w_k \\
\text{subject to} \quad g + \nabla g^T \Delta w_k = 0 \\
\quad \quad h + \nabla h^T \Delta w_k \geq 0
\]
Nonlinear Programming and SQP

Without Inequalities

- Linearize the KKT system
  \[
  \begin{bmatrix}
  \nabla^2_w \mathcal{L} & \nabla g \\
  \nabla g^T & 0
  \end{bmatrix}
  \begin{bmatrix}
  \Delta w_k \\
  \lambda_{k+1}
  \end{bmatrix}
  =
  \begin{bmatrix}
  \nabla f \\
  g
  \end{bmatrix}
  \]

- Solve the linear system
- Corresponding QP

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \Delta w_k^T \nabla^2_w \mathcal{L} \Delta w_k + \nabla f^T \Delta w_k \\
\text{subject to} & \quad g + \nabla g^T \Delta w_k = 0
\end{align*}
\]

With Inequalities
Without Inequalities

- Linearize the KKT system

\[
\begin{bmatrix}
\nabla^2_w \mathcal{L} & \nabla g \\
\nabla g^T & 0
\end{bmatrix}
\begin{bmatrix}
\Delta w_k \\
\lambda_{k+1}
\end{bmatrix}
= -
\begin{bmatrix}
\nabla f \\
g
\end{bmatrix}
\]

- Solve the linear system
- Corresponding QP

\[
\text{minimize } \frac{1}{2} \Delta w_k^T \nabla^2_w \mathcal{L} \Delta w_k + \nabla f^T \Delta w_k
\]

subject to \( g + \nabla g^T \Delta w_k = 0 \)

With Inequalities

- The last 3 KKT conditions are nonsmooth.
Without Inequalities

- Linearize the KKT system

\[
\begin{bmatrix}
\nabla^2_w \mathcal{L} & \nabla g \\
\nabla g^T & 0
\end{bmatrix}
\begin{bmatrix}
\Delta w_k \\
\lambda_{k+1}
\end{bmatrix}
= -
\begin{bmatrix}
\nabla f \\
g
\end{bmatrix}
\]

- Solve the linear system
- Corresponding QP

\[
\begin{align*}
\min_{\Delta w_k} & \quad \frac{1}{2} \Delta w_k^T \nabla^2_w \mathcal{L} \Delta w_k + \nabla f^T \Delta w_k \\
\text{subject to} & \quad g + \nabla g^T \Delta w_k = 0
\end{align*}
\]

With Inequalities

- The last 3 KKT conditions are nonsmooth.
- At each iteration solve the QP

\[
\begin{align*}
\min_{\Delta w_k} & \quad \frac{1}{2} \Delta w_k^T \nabla^2_w \mathcal{L} \Delta w_k + \nabla f^T \Delta w_k \\
\text{subject to} & \quad g + \nabla g^T \Delta w_k = 0 \\
& \quad h + \nabla h^T \Delta w_k \geq 0
\end{align*}
\]
QP solution

“QP is almost a technology”, S. Boyd
QP solution

"QP is almost a technology", S. Boyd

Convex QP:

- No inequalities: solve a linear system
QP solution

“QP is almost a technology”, S. Boyd

Convex QP:

- **No inequalities**: solve a linear system
- **Inequalities**: interior point or **active set** method
  
  **Active set** algorithm
  
  - guess active constr.
  - solve linear system
  - add/remove constr.
QP solution

“QP is almost a technology”, S. Boyd

Convex QP:

- No inequalities: solve a linear system
- Inequalities: interior point or active set method

Active set: algorithm
- guess active constr.
- solve linear system
- add/remove constr.

properties
- can be warm started
- extremely fast with good initial guess

Many reliable QP solvers available:
- qpOASES
- FORCES
- quadprog
- many others
**QP solution**

“QP is almost a technology”, S. Boyd

**Convex QP:**
- **No inequalities:** solve a linear system
- **Inequalities:** interior point or **active set** method

  Active set: algorithm
  - guess active constr.
  - solve linear system
  - add/remove constr.

  properties
  - can be warm started
  - extremely fast with good initial guess

**Nonconvex QP:** **NP-hard** problem
QP solution

“QP is almost a technology”, S. Boyd

Convex QP:
- **No inequalities**: solve a linear system
- **Inequalities**: interior point or **active set** method

Active set: algorithm
- guess active constr.
- solve linear system
- add/remove constr.

properties
- can be warm started
- extremely fast with good initial guess

Nonconvex QP: **NP-hard** problem

Many **reliable QP solvers** available:
- qpOASES
- FORCES
- quadprog
- many others
**SQP method in a nutshell**

**NMPC at time** $i$

$$\begin{align*}
\min_{w} & \quad f(w) \\
\text{s.t.} & \quad g(w) \\
& \quad h(w) \geq 0
\end{align*}$$

**Iterative procedure:**
**SQP method in a nutshell**

<table>
<thead>
<tr>
<th>NMPC at time $i$</th>
<th>Quadratic Problem Approximation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\min_w f(w)$</td>
<td>$\min_{\Delta w} \frac{1}{2} \Delta w$</td>
</tr>
<tr>
<td>s.t. $g(w)$</td>
<td>$\Delta w + \Delta w$</td>
</tr>
<tr>
<td>$h(w) \geq 0$</td>
<td>s.t. $+ \Delta w = 0$</td>
</tr>
<tr>
<td></td>
<td>$+ \Delta w \geq 0,$</td>
</tr>
</tbody>
</table>

**Iterative procedure:**

1. Given current guess $w_k, \lambda_k, \mu_k$

**SQP method in a nutshell**

Quadratic Problem Approximation

1. **QP (for a given $s, u$)**

   - $\min_{\Delta w} \frac{1}{2} \Delta w$
   - $\Delta w + \Delta w$
   - s.t. $+ \Delta w = 0$
   - $+ \Delta w \geq 0,$
### SQP method in a nutshell

**NMPC at time \( i \)**

\[
\begin{align*}
\min_w & \quad f(w) \\
\text{s.t.} & \quad g(w) \\
& \quad h(w) \geq 0
\end{align*}
\]

**Quadratic Problem Approximation**

**QP** (for a given \( s, u \))

\[
\begin{align*}
\min_{\Delta w} & \quad \frac{1}{2} \Delta w \, B(w_k) \, \Delta w + \nabla f(w_k)^T \Delta w \\
\text{s.t.} & \quad g(w_k) + \nabla g(w_k)^T \, \Delta w = 0 \\
& \quad h(w_k) + \nabla h(w_k)^T \, \Delta w \geq 0,
\end{align*}
\]

**Iterative procedure:**

1. **Given current guess** \( w_k, \lambda_k, \mu_k \)
2. **Linearize** at \( w_k, \lambda_k, \mu_k \): need 2\(^{nd}\) order derivatives for \( B(w_k) \)
### SQP method in a nutshell

**NMPC at time $i$**

$$ \begin{align*} & \min_w f(w) \\
& \text{s.t. } g(w) \\
& \quad h(w) \geq 0 \end{align*} $$

**QP (for a given $s, u$)**

$$ \begin{align*} & \min_{\Delta w} \frac{1}{2} \Delta w \, B(w_k) \, \Delta w + \nabla f(w_k)^T \Delta w \\
& \text{s.t. } g(w_k) + \nabla g(w_k)^T \Delta w = 0 \\
& \quad h(w_k) + \nabla h(w_k)^T \Delta w \geq 0, \end{align*} $$

### Iterative procedure:

1. Given current guess $w_k$, $\lambda_k$, $\mu_k$

2. **Linearize** at $w_k$, $\lambda_k$, $\mu_k$: need 2nd order derivatives for $B(w_k)$

3. Make sure Hessian $B(w_k) \succ 0$: avoid negative curvature
## SQP method in a nutshell

**NMPC at time \( i \)**

\[
\begin{align*}
\min_w & \quad f(w) \\
\text{s.t.} & \quad g(w) \\
& \quad h(w) \geq 0
\end{align*}
\]

**QP (for a given \( s, u \))**

\[
\begin{align*}
\min_{\Delta w} & \quad \frac{1}{2} \Delta w \ B(w_k) \ \Delta w + \nabla f(w_k)^T \Delta w \\
\text{s.t.} & \quad g(w_k) + \nabla g(w_k)^T \ \Delta w = 0 \\
& \quad h(w_k) + \nabla h(w_k)^T \ \Delta w \geq 0,
\end{align*}
\]

### Iterative procedure:

1. Given current guess \( w_k, \lambda_k, \mu_k \)
2. **Linearize** at \( w_k, \lambda_k, \mu_k \): need 2\(^{nd} \) order derivatives for \( B(w_k) \)
3. Make sure Hessian \( B(w_k) \succeq 0 \): avoid negative curvature
4. Solve QP
SQP method in a nutshell

**NMPC at time $i$**

\[
\begin{align*}
\min_{w} & \quad f(w) \\
\text{s.t.} & \quad g(w) \\
& \quad h(w) \geq 0
\end{align*}
\]

**Quadratic Problem Approximation**

**QP (for a given $s$, $u$)**

\[
\begin{align*}
\min_{\Delta w} & \quad \frac{1}{2} \Delta w \, B(w_k) \, \Delta w + \nabla f(w_k)^T \Delta w \\
\text{s.t.} & \quad g(w_k) + \nabla g(w_k)^T \Delta w = 0 \\
& \quad h(w_k) + \nabla h(w_k)^T \Delta w \geq 0,
\end{align*}
\]

Iterative procedure:

1. Given current guess $w_k$, $\lambda_k$, $\mu_k$
2. **Linearize** at $w_k$, $\lambda_k$, $\mu_k$: need 2\textsuperscript{nd} order derivatives for $B(w_k)$
3. Make sure Hessian $B(w_k) \succ 0$: avoid negative curvature
4. Solve QP
5. Globalization (e.g. line-search): **ensure descent**, stepsize $\alpha \in (0, 1]$
SQP method in a nutshell

NMPC at time $i$

\[
\begin{align*}
\min_w & \quad f(w) \\
\text{s.t.} & \quad g(w) \\
& \quad h(w) \geq 0
\end{align*}
\]

Iterative procedure:

1. Given current guess $w_k$, $\lambda_k$, $\mu_k$
2. **Linearize** at $w_k$, $\lambda_k$, $\mu_k$: need 2\textsuperscript{nd} order derivatives for $B(w_k)$
3. Make sure Hessian $B(w_k) \succ 0$: avoid negative curvature
4. Solve QP
5. Globalization (e.g. line-search): **ensure descent**, stepsize $\alpha \in (0, 1]$

6. Update

\[
\begin{bmatrix}
w_{k+1} \\
\lambda_{k+1} \\
\mu_{k+1}
\end{bmatrix} = \begin{bmatrix}
w_k \\
\lambda_k \\
\mu_k
\end{bmatrix} + \alpha \begin{bmatrix}
\Delta w \\
\Delta \lambda \\
\Delta \mu
\end{bmatrix}
\]

and iterate
1. Nonlinear Programming and SQP

Specific Structure of $f(w)$

$$\begin{align*}
\text{minimize} & \quad \frac{1}{2} \| F(w) \|_2^2 \\
\text{subject to} & \quad g(w) = 0 \\
& \quad h(w) \geq 0 \quad (2)
\end{align*}$$
Specific Structure of $f(w)$

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \| F(w) \|_2^2 \\
\text{subject to} & \quad g(w) = 0 \\
& \quad h(w) \geq 0
\end{align*}
\]

(2)

Gauss-Newton Hessian Approximation

Linearize inside the norm to obtain

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \| F(w_0) + J(w_0) \Delta w \|_2^2 \\
\text{subject to} & \quad g(w_0) + \nabla g(w_0)^T \Delta w = 0 \\
& \quad h(w_0) + \nabla h(w_0)^T \Delta w \geq 0
\end{align*}
\]

where $J(w_0) = \nabla F(w_0)^T$. 
Why Does it perform well?

- **Exact Hessian:**
  \[
  \nabla_w^2 \mathcal{L} = \nabla^2 f - \sum \lambda_i \nabla^2 g_i - \sum \mu_i \nabla^2 h_i \\
  = J^T J + \sum F_i \nabla^2 F_i - \sum \lambda_i \nabla^2 g_i - \sum \mu_i \nabla^2 h_i
  \]

- **Gauss-Newton Hessian:**
  \[
  \nabla_w^2 \mathcal{L} \approx J^T J
  \]

**No need for**
- 2nd order derivatives
- Lagrange multipliers

When Does it perform well?
- \( \|F\| \) small: good fit
- \( \nabla^2 F_i \) small: residuals \( F \) nearly linear
- \( \|\lambda\| \) and \( \|\mu\| \) small: true when \( \|F\| \) small
Wide Range of Applications

- System identification:

  \[ \min_p \| y(p) - y \|_S^2 \]

- Model Predictive Control:

  \[ \min_{x,u} \| x - x^r \|_Q^2 + \| u - u^r \|_R^2 \]

- Moving Horizon Estimation:

  \[ \min_{x,u} \| y(x, u) - y \|_S^2 \]