

Overview

Quadratic programming (QP) is a very useful form of convex optimization. It consists of a quadratic objective with linear constraints. Because it is convex, if the constraint set is feasible, the global minimum is guaranteed to be found. Further, QPs can be solved very quickly and are amenable for real-time processing.

Many problems in optimal control can be formulated with a quadratic objective and linear constraints. In particular, (linear) model predictive control (MPC) can be formulated as a QP which can be efficiently solved using existing solvers. In this report, we give an overview of quadratic programming and the qpOASES¹ solver from COIN-OR. Additionally, we give examples of MPC, where QPs naturally arise.

QP Formulation

A quadratic programming problem with n variables and m constraints can be formulated as

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && \frac{1}{2} \mathbf{x}^\top Q \mathbf{x} + \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && A \mathbf{x} \leq \mathbf{b}, \end{aligned} \quad (1)$$

where $\mathbf{x} \in \mathbf{R}^n$ is the optimization variable and $Q \in \mathbf{S}_+^n$ (symmetric, p.s.d.), $\mathbf{c} \in \mathbf{R}^n$, $A \in \mathbf{R}^{m \times n}$, and $\mathbf{b} \in \mathbf{R}^m$ are the problem data. Note: if we were maximizing, we would want $Q \in \mathbf{S}_-^n$ (n.s.d.) or \mathbf{S}_{--}^n (n.d.).

Least Squares

A common problem is to fit data to some model, i.e., regression. Given a batch of measurements $\mathbf{y} \in \mathbf{R}^m$, we wish to find the parameters $\mathbf{x} \in \mathbf{R}^n$ of some model $f: \mathbf{R}^n \rightarrow \mathbf{R}^m$ so that $f(\mathbf{x}) = \mathbf{y}$. In linear regression, $f(\mathbf{x}) = A\mathbf{x}$ with $A \in \mathbf{R}^{m \times n}$. When the columns of A are linearly independent and $m > n$, there are “more equations than unknowns”, implying that there is no unique solution² \mathbf{x}^* . We call the system of equations *overdetermined*. A common approach is then to find the “best fit”—the \mathbf{x}^* that minimizes the squared norm of the residual $\|\mathbf{r}\|_2^2 = \|A\mathbf{x} - \mathbf{y}\|_2^2$. This is known as the least squares solution.

There are two main methods of derivation: one uses the principle of orthogonality from linear algebra and the other is the usual calculus method of optimization. Here, we will leverage the calculus method to solve

$$\min_{\mathbf{x}} \quad \frac{1}{2} \|\mathbf{r}\|_2^2 = \frac{1}{2} \mathbf{r}^\top \mathbf{r} = \frac{1}{2} (A\mathbf{x} - \mathbf{y})^\top (A\mathbf{x} - \mathbf{y}) = \frac{1}{2} \mathbf{x}^\top A^\top A \mathbf{x} - \mathbf{y}^\top A \mathbf{x} + \frac{1}{2} \mathbf{y}^\top \mathbf{y}. \quad (2)$$

By setting the Jacobian³ of the residual to zero, we can solve directly for the minimizer:

$$\begin{aligned} \frac{\partial}{\partial \mathbf{x}} \frac{1}{2} \|\mathbf{r}\|_2^2 &= A^\top A \mathbf{x}^* - A^\top \mathbf{y} = 0 \\ A^\top A \mathbf{x}^* &= A^\top \mathbf{y} \\ \mathbf{x}^* &= (A^\top A)^{-1} A^\top \mathbf{y} \\ \mathbf{x}^* &= A^\dagger \mathbf{y}, \end{aligned} \quad (3)$$

where A^\dagger is the left Moore-Penrose pseudoinverse.

Note from equation (2) that minimizing the residual in the least-squares sense actually involves the quadratic form

$$J(\mathbf{x}) = \frac{1}{2} \mathbf{x}^\top Q \mathbf{x} + \mathbf{c}^\top \mathbf{x}, \quad (4)$$

where the $\mathbf{y}^\top \mathbf{y}$ term is constant w.r.t \mathbf{x} and is ignored, $\mathbf{c} = -A^\top \mathbf{y}$ and $Q = A^\top A$ is symmetric positive-definite by construction⁴. Hence, the least squares problem (2) can be formulated as an unconstrained QP (1). Even constrained least squares problems can be reformulated as QPs⁵. However, the converse is not true: since a standard QP only requires that $Q \in \mathbf{S}_+^n$, there may not be a Cholesky decomposition into an invertible Gram matrix $A^\top A$.

¹<https://projects.coin-or.org/qpOASES>

²As opposed to the rows of A being l.i. with $m < n$, which implies that there are infinitely many solutions (min norm).

³The Derivative of a Quadratic Form

⁴The product $A^\top A$ is a Gram matrix, which is made up of inner products. Therefore, all Gram matrices are positive semidefinite. The Gram matrix is invertible if and only if the set of vectors are linearly independent. Since we required the columns of A to be linearly independent, we see that $\det(A^\top A) \neq 0$.

⁵SE.Math Post

Graphical Examples⁶

Consider the following QP

$$\begin{aligned}
 &\underset{x_1, x_2}{\text{minimize}} && (x_1 - 2)^2 + (x_2 - 2)^2 \\
 &\text{subject to} && 2x_1 + 4x_2 \leq 28 \\
 &&& 5x_1 + 5x_2 \leq 50 \\
 &&& x_1 \leq 8 \\
 &&& x_2 \leq 8 \\
 &&& x_1 \geq 0 \\
 &&& x_2 \geq 0.
 \end{aligned} \tag{5}$$

As this QP is a function of two variables, it can be solved graphically by plotting the contour lines of the quadratic form along with the linear constraints. In linear programming, the optimum is always found on the boundary of the feasible set. In QPs, the optimal solution may be in the interior of the feasible set (as is the case for problem (5)) or it may be on the boundary.

Optimality Conditions for Unconstrained Optimization

Recall the following result from single-variable calculus:

Theorem 1 (Rudin 5.8). *Let f be a real-valued function defined on $[a, b]$. If f has a local minimum at a point $x \in (a, b)$ and $f'(x)$ exists, then $f'(x) = 0$.*

Proof. Assume x is a local minimum of f . Then there exists a $\delta > 0$ such that $f(p) \leq f(x)$ for all p with $|p - x| < \delta$. Given this δ , we have that $a < x - \delta < x < x + \delta < b$.

If $x - \delta < p < x$, then

$$\frac{f(p) - f(x)}{p - x} \leq 0.$$

As $p \rightarrow x$, we see that $f'(x) \leq 0$.

If $x < p < x + \delta$, then

$$\frac{f(p) - f(x)}{p - x} \geq 0,$$

and as $p \rightarrow x$, we see that $f'(x) \geq 0$. Hence, $f'(x) = 0$. □

We claim without proof that this condition is also true for multivariate calculus. In fact, we have already used this result when solving for the least squares solution in equation (3). We call this condition the *first-order necessary condition* for optimality.

However, this condition is not sufficient to determine if \mathbf{x}^* is a minimizer, maximizer, or stationary point. So how were we guaranteed that this was the correct \mathbf{x}^* in the least squares discussion above? In fact, how could we claim in the introduction that if a convex QP is feasible its global minimizer can always be found? To answer these questions, we must use the *second-order sufficient condition* for optimality.

Again using single-variable calculus as a reference, recall that the second derivative of a function can reveal if a stationary point is a minimizer, maximizer, or neither. For a single-variable function f , the stationary point x^* is characterized as

$$\begin{aligned}
 &\text{a minimizer} && \text{if } f''(x^*) > 0, \\
 &\text{a maximizer} && \text{if } f''(x^*) < 0, \\
 &\text{an inflection point} && \text{if } f''(x^*) = 0.
 \end{aligned}$$

The second derivative of the multivariable function is called the Hessian. Without proof, we claim that it is sufficient to show that if the Hessian is positive definite, then a stationary point \mathbf{x}^* is the unique global minimizer. If instead the Hessian is positive semidefinite, then there is a valley of solutions. The relationship between the Hessian of a function and the nature of a stationary point \mathbf{x}^* is summarized in Table 1.

Therefore, to answer the question posed about whether or not the solution \mathbf{x}^* found from least squares is guaranteed to be the global minimizer, we calculate the Hessian of equation (2) as

$$\frac{\partial^2}{\partial \mathbf{x}^2} \left[\frac{1}{2} \mathbf{x}^\top Q \mathbf{x} + c^\top \mathbf{x} \right] = Q, \tag{6}$$

⁶UC Berkeley, CE 191

Table 1: Hessian and the nature of stationary point \mathbf{x}^*

Hessian Matrix Definiteness	Nature of \mathbf{x}^*
positive definite	minimizer
positive semidefinite	valley
negative definite	maximizer
negative semidefinite	ridge
indefinite	saddle point

and use the second-order sufficient condition. Since we showed that $Q \succ 0$ by construction, we see that \mathbf{x}^* is in fact the unique global minimizer⁷.

Finally, we note that this discussion about first-order necessary and second-order sufficient conditions has been for the unconstrained optimization case. When constraints are incorporated using Lagrange multipliers, new (similar) second-order sufficient conditions arise.

Introduction to qpOASES

Having covered some theoretical aspects of QPs, we turn our attention to using qpOASES to solve them. The qpOASES solver expects QPs in the following standard form:

$$\begin{aligned}
 & \underset{\mathbf{x}}{\text{minimize}} && \frac{1}{2} \mathbf{x}^\top H \mathbf{x} + \mathbf{x}^\top \mathbf{g}(w_0) \\
 & \text{subject to} && \text{lb}A(w_0) \leq A\mathbf{x} \leq \text{ub}A(w_0) \\
 & && \text{lb}(w_0) \leq \mathbf{x} \leq \text{ub}(w_0)
 \end{aligned} \tag{7}$$

where $\mathbf{x} \in \mathbf{R}^n$ is the optimization variable, $H \in \mathbf{S}_+^n$ is the Hessian, $\mathbf{g} \in \mathbf{R}^n$ is the gradient vector, $A \in \mathbf{R}^{m \times n}$, and the (element-wise) upper and lower bounds are of appropriate dimensions. Notice that the gradient vector and the upper and lower bounds are affine functions of the parameter w_0 .

Example

Rewriting problem (5) into qpOASES standard form, we have that

$$H = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}; \quad \mathbf{g}^\top = [-4 \quad -4]; \quad A = \begin{bmatrix} 2 & 4 \\ 5 & 5 \end{bmatrix}; \quad \text{lb}A = \begin{bmatrix} -\infty \\ -\infty \end{bmatrix}; \quad \text{ub}A = \begin{bmatrix} 28 \\ 50 \end{bmatrix}; \quad \text{lb} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}; \quad \text{ub} = \begin{bmatrix} 8 \\ 8 \end{bmatrix}.$$

The solution to this QP is easily seen to be $\mathbf{x}^* = [2 \quad 2]$, with no active constraints. We can check this in MATLAB with the following call to `quadprog`:

```
>> x = quadprog(H,g,A,ubA,[],[],lb,ub)
x =
    2
    2
```

Using qpOASES, this QP can be solved using the following code:

```
#include <iostream>
#include <qpOASES.hpp>

int main()
{
    constexpr int n = 2;
    constexpr int m = 2;

    // problem data
    qpOASES::real_t H[n*n] = {2.0, 0.0, 0.0, 2.0};
    qpOASES::real_t g[n] = {-4.0, -4.0};
    qpOASES::real_t A[m*n] = {2.0, 4.0, 5.0, 5.0};
    qpOASES::real_t * lbA = nullptr; // no lower bound
```

⁷If $Q \succeq 0$ and the constraint set is feasible there will be multiple solutions, SE.Math.

```

qpOASES::real_t ubA[m] = {28.0, 50.0};
qpOASES::real_t lb[n] = {0.0, 0.0};
qpOASES::real_t ub[n] = {8.0, 8.0};

qpOASES::QProblem P(n, m);

// solve QP
int nWSR = 10; // max num "working set recalculations"
P.init(H, g, A, lb, ub, lbA, ubA, nWSR);

qpOASES::real_t xOpt[n];
P.getPrimalSolution(xOpt);
std::cout << "xOpt:" << std::endl;
std::cout << "\t" << xOpt[0] << std::endl << "\t" << xOpt[1] << std::endl;
std::cout << "objVal: " << P.getObjVal() << std::endl;

return 0;
}

```

Which has the following output:

```

##### qpOASES -- QP NO. 1 #####

  Iter | StepLength | Info | nFX | nAC
-----+-----+-----+-----+-----
    0 | 1.884058e-01 | REM BND 0 | 1 | 0
    1 | 1.428571e-02 | REM BND 1 | 0 | 0
    2 | 1.000000e+00 | QP SOLVED | 0 | 0
xOpt:
  2
  2
objVal: -8

```