# Primal-dual Interior-Point Methods with Application to Optimal Control

Thiago da Cunha Vasco

December 8, 2024

**Abstract**

For my CPSC 536M project I decided to focus on primal-dual interior-point methods and their application to solving both linear and nonlinear optimal control problems. First I formulate the general optimization problem we will encounter in the rest of the report. In the following sections, I go over the intuition and theory required to formulate primal-dual interior-point methods, and then apply these methods to the more specific problems of quadratic programming (QP) and nonlinear programming (NLP). For both QP and NLP, I provide examples in the context of optimal control. The code for this project is available at https://github.com/Thiagodcv/interior-point-optimal-control.git.

## 1 Introduction

In this report, we assume that all functions we're working with are $C^2$, and are mappings between finite-valued Euclidean spaces of potentially different dimension. We define the general optimization problem here. Let $z \in \mathbb{R}^{n_z}$ be the decision variable we want to optimize, where $n_z$ is its dimension. Further, let $f_0 : \mathbb{R}^{n_z} \to \mathbb{R}$ be the objective function we want to minimize, subject to inequality constraints set by functions $f_i : \mathbb{R}^{n_z} \to \mathbb{R}$ for $i \in \mathcal{I}$, and equality constraints set by functions $h_i : \mathbb{R}^{n_z} \to \mathbb{R}$ for $i \in \mathcal{E}$. The problem we want to solve can be formulated as

$$\begin{aligned} \min_{z} \quad & f_0(z) \\ \text{subject to} \quad & f_i(z) \leq 0, \ i \in [n_{\mathcal{I}}] \\ & h_i(z) = 0, \ i \in [n_{\mathcal{E}}]. \end{aligned} \tag{1}$$

where $n_{\mathcal{I}} := |\mathcal{I}|$ and $n_{\mathcal{E}} := |\mathcal{E}|$ are defined to be the number of the inequality and equality constraints respectively. Precisely how we go about trying to solve (1) will depend on further assumptions we make about the functions defined above. However, common to all primal-dual interior-point methods is the use of KKT conditions and central path. We will cover both in the following subsections.

### 1.1 Duality

#### 1.1.1 Primal and Dual Problems

Before we can discuss KKT conditions, we must first discuss duality. To start off, note how one can reformulate problem (1) as the minimization problem

$$\min_{z} \sup_{\lambda \geq 0, \ \nu} \left\{ f_0(z) + \sum_{i=1}^{n_{\mathcal{I}}} \lambda_i f_i(z) + \sum_{j=1}^{n_{\mathcal{E}}} \nu_j h_j(z) \right\}. \tag{2}$$

Note how if we pick $z$ such that some inequality constraint isn't satisfied (i.e., $f_i(z) > 0, i \in [n_{\mathcal{I}}]$), or such that some equality constraint isn't satisfied (i.e., $h_i(z) \neq 0, i \in [n_{\mathcal{E}}]$), then the objective goes to infinity. We define the Lagrangian to be equal to

$$\mathcal{L}(z, \lambda, \nu) = f_0(z) + \sum_{i=1}^{n_{\mathcal{I}}} \lambda_i f_i(z) + \sum_{j=1}^{n_{\mathcal{E}}} \nu_j h_j(z). \tag{3}$$

We call (2) the *primal* problem. Conversely, we have that the *dual* problem associated with $\mathcal{L}$ is

$$\begin{aligned} &\max_{\lambda,\ \nu} \ \inf_{z} \mathcal{L}(z, \lambda, \nu) \\ &\text{subject to } \lambda \geq 0. \end{aligned} \tag{4}$$

The primal and dual problems can be given a game-theoretic or economic interpretation which we won't spent too much time trying to formalize. To put it briefly, in the primal problem, the agent who is controlling $z$ gets to go first. He is trying to minimize the maximum cost knowing that after he inserts his choice of $\bar{z}$, his adversary will then choose $\lambda \geq 0$ and $\nu$ so that $\mathcal{L}(\bar{z}, \lambda, \nu)$ is maximized. Conversely in the dual problem, the agent who is controlling $(\lambda, \nu)$ gets to go first. She is trying to maximize the minimum cost knowing that after she inserts her choice of $(\bar{\lambda}, \bar{\nu})$, her adversary will then choose $z$ so that $\mathcal{L}(z, \bar{\lambda}, \bar{\nu})$ is minimized.

Let $p^*$ be the optimal value of primal problem (2), and let $d^*$ be the optimal value of dual problem (4). In general, we have that

$$d^* \leq p^*. \tag{5}$$

This property is called *weak duality*. Furthermore, we refer to the value $p^* - d^*$ as the *optimal duality gap*, as it represents the difference between the optimal value of the primal problem, and the optimal value of the dual problem. Using our game-theoretic interpretation, we can interpret a positive optimal duality gap to mean that agents are worse off being the first to choose their variable. For some problems, we may have that

$$d^* = p^* \tag{6}$$

holds true. Property (6) is called *strong duality*. In this case, there is no disadvantage for either agent to move first. We will now discuss under what condition strong duality holds.

### 1.1.2 Conditions for Strong Duality

There are several ways one can ensure strong duality; we will cover only the conditions necessary for the scope of this project. First, we must discuss convexity. Problem (1) is *convex* if $f_i$ for $i \in \{0\} \cup [n_{\mathcal{I}}]$ are convex, and $h_i$ for $i \in [n_{\mathcal{E}}]$ are affine. Note how if this condition holds, (3) is also convex in $z$. Because the equality constraints are affine, they can simply be expressed as $Cz = b$, where $C \in \mathbb{R}^{n_{\mathcal{E}} \times n_z}$ is an underdetermined matrix with full row rank.

The other condition we must discuss is *Slater's condition*. Let $\mathcal{D}$ be the intersection of the domains of all functions in (1). Slater's condition is satisfied if there exists $z \in \text{relint}(\mathcal{D})$ such that $f_i(z) < 0$ for all $i \in [n_{\mathcal{I}}]$ and $Cz = b$.

**Theorem 1.1.** *Suppose (1) is convex, and that Slater's condition holds. Then strong duality holds.*

When the first $k$ inequality constraints are affine, strong duality also holds under a "refined" version of Slater's condition: there exists $z \in \text{relint}(\mathcal{D})$ such that

$$f_i(z) \leq 0, \quad i = 1, \ldots, k, \qquad f_i(x) < 0, \quad i = k+1, \ldots, [n_{\mathcal{I}}], \qquad Cz = b. \tag{7}$$

Note that when all of the inequality functions are affine, and when $D$ is an open set, the refined Slater's condition simply reduces to feasibility.

### 1.1.3 Implications of Strong Duality (Complementary Slackness)

Let $z^*$ be a solution to a primal problem (1), and $(\lambda^*, \nu^*)$ for $\lambda^* \geq 0$ be a solution to its associated dual problem (4). We say that $z^*$ and $(\lambda^*, \nu^*)$ are primal and dual optimal respectively. Furthermore, suppose strong duality holds. Then

$$\begin{aligned} f_0(z^*) &= g(\lambda^*, \nu^*) \\ &= \inf_z \left\{ f_0(z) + \sum_i \lambda_i^* f_i(z) + \sum_j \nu_j^* h_j(z) \right\} \\ &\leq f_0(z^*) + \sum_i \lambda_i^* f_i(z^*) + \sum_j \nu_j^* h_j(z^*) \\ &\leq f_0(z^*). \end{aligned} \tag{8}$$

Because $h_j(z^*) = 0$, this implies $\sum_i \lambda_i^* f_i(z^*) = 0$. Since all terms in this sum are less than or equal to zero, this implies $\lambda_i^* f_i(z^*) = 0$ for $i \in [n_{\mathcal{I}}]$. Furthermore, because the last inequality in (8) is actually an equality, this implies that $z^*$ is a minimizer of $\mathcal{L}(z, \lambda^*, \nu^*)$.

### 1.1.4 KKT Optimality Conditions

Having discussed duality and some of its basic results, we can now discuss the KKT conditions and their integral role in solving optimization problems of the form (1).

**KKT Necessary Conditions**
Suppose $z^*$ and $(\lambda^*, \nu^*)$ for $\lambda^* \geq 0$ are primal and dual optimal respectively, with zero duality gap. Recall from section 1.1.3 that this implies $z^*$ minimizes $\mathcal{L}(z, \lambda^*, \nu^*)$. By first order conditions we therefore have that $\nabla_z \mathcal{L}(z^*, \lambda^*, \nu^*) = 0$. Another result from section 1.1.3 is that $\lambda_i^* f_i(z^*) = 0$ for $i \in [n_{\mathcal{I}}]$. We therefore have that any primal-dual optimal $z^*$ and $(\lambda^*, \nu^*)$ with zero duality gap satisfies

$$\begin{aligned} f_i(z^*) &\leq 0 \qquad i \in [n_{\mathcal{I}}] \\ h_i(z^*) &= 0 \qquad i \in [n_{\mathcal{E}}] \\ \lambda_i^* &\geq 0 \qquad i \in [n_{\mathcal{I}}] \\ \lambda_i^* f_i(z^*) &= 0 \qquad i \in [n_{\mathcal{I}}] \\ \nabla f_0(z^*) + \sum_{i=1}^{n_{\mathcal{I}}} \lambda_i^* \nabla f_i(z^*) + \sum_{i=1}^{n_{\mathcal{E}}} \nu_i^* \nabla h_i(z^*) &= 0, \end{aligned} \tag{9}$$

which is referred to as the *Karush-Kuhn-Tucker* (KKT) conditions. Recall that the condition for zero duality gap can be easily satisfied using Slater's condition in many cases.

**KKT Sufficient Conditions**
Suppose problem (1) is convex (i.e., $f_i$ are convex and $h_i$ are affine). Then if points $\tilde{z}$, $\tilde{\lambda}$, and $\tilde{\nu}$ satisfy the KKT conditions

$$\begin{aligned} f_i(\tilde{z}) &\leq 0 \qquad i \in [n_{\mathcal{I}}] \\ h_i(\tilde{z}) &= 0 \qquad i \in [n_{\mathcal{E}}] \\ \tilde{\lambda}_i &\geq 0 \qquad i \in [n_{\mathcal{I}}] \\ \tilde{\lambda}_i f_i(\tilde{z}) &= 0 \qquad i \in [n_{\mathcal{I}}] \\ \nabla f_0(\tilde{z}) + \sum_{i=1}^{n_{\mathcal{I}}} \tilde{\lambda}_i \nabla f_i(\tilde{z}) + \sum_{i=1}^{n_{\mathcal{E}}} \tilde{\nu}_i \nabla h_i(\tilde{z}) &= 0, \end{aligned} \tag{10}$$

then $\tilde{z}$ and $(\tilde{\lambda}, \tilde{\nu})$ are primal and dual optimal with zero duality gap. The proof of this is actually quite simple. Note how with our convexity assumption, $\mathcal{L}(z, \tilde{\lambda}, \tilde{\nu})$ is convex in $z$. Because the last KKT condition says that $\tilde{z}$ is a stationary point with respect to $\mathcal{L}(z, \tilde{\lambda}, \tilde{\nu})$, we then have that $\tilde{z}$ is actually its minimizer. So then

$$\begin{aligned} g(\tilde{\lambda}, \tilde{\nu}) &= \inf_z \mathcal{L}(z, \tilde{\lambda}, \tilde{\nu}) \\ &= \mathcal{L}(\tilde{z}, \tilde{\lambda}, \tilde{\nu}) \\ &= f_0(\tilde{z}) + \sum_{i=1}^{n_{\mathcal{I}}} \lambda_i f_i(\tilde{z}) + \sum_{i=1}^{n_{\mathcal{E}}} \nu_i h_i(\tilde{z}) \\ &= f_0(\tilde{z}), \end{aligned} \tag{11}$$

and because in general $g(\tilde{\lambda}, \tilde{\nu}) \leq d^* \leq p^* \leq f_0(\tilde{z})$, we have that optimality and strong duality both hold.

Finally, it is worth highlighting that for the class of all problems of the form (1) which are convex and which satisfy Slater's condition, $\tilde{z}$ and $(\tilde{\lambda}, \tilde{\nu})$ are primal and dual optimal if and only if the KKT conditions hold at those points. The KKT conditions are important as they serve as the foundation of primal-dual interior point methods, and as well as many other methods of optimization. Now that we've covered the most important and relevant aspects of duality, we will slowly make our way towards more practical matters in constructing a solver.

## 1.2 The Central Path

In general, we are trying to solve a problem of the form (1), which contains both equality and inequality constraints. This section will review how (1) can be reformulated to only contain equality constraints. This is desirable as then the problem can be solved using linearly-constrained Newton's method. In this section, we will assume that (1) is a convex problem, and hence, $h(z) := Cz - b$.

We first observe how (1) is equivalent to solving

$$\begin{aligned} &\min_z f_0(z) + \sum_{i=1}^{n_{\mathcal{I}}} I_-(f_i(z)) \\ &\text{such that } Cz - b = 0, \end{aligned} \tag{12}$$

where $I_- : \mathbb{R} \to \mathbb{R}$ is an indicator function defined as

$$I_-(u) := \begin{cases} 0 & u \leq 0 \\ \infty & u > 0. \end{cases} \tag{13}$$

The obvious problem with this reformulation is the fact that $I_-$ has a zero derivative everywhere in $(-\infty, 0)$, and has an undefined derivative everywhere else. This can be remedied by approximating $I_-$ with another function $\hat{I}_- : (-\infty, 0) \to \mathbb{R}$ defined as

$$\hat{I}_-(u) := -\mu \log(-u) \tag{14}$$

for $\mu > 0$. Note how $\lim_{u \to 0^-} \hat{I}_-(u) = \infty$, whereas $\lim_{u \to 0^-} I_-(u) = 0$. This means that this modified function punishes for getting too close to constraint boundaries, which is useful from an optimization perspective. Furthermore, as $\mu$ approaches 0 we have that $\hat{I}_-$ converges to $I_-$ pointwise; this means that we can get a close approximation to the original problem by setting $\mu$ small. Lastly, we have that $\hat{I}_-$ is convex.

We define the *logarithmic barrier* function to be

$$\phi(z) := -\sum_{i=1}^{n_\mathcal{I}} \log(-f_i(z)). \tag{15}$$

It will be useful for later to note that

$$\nabla \phi(z) = \sum_{i=1}^{n_\mathcal{I}} \frac{1}{-f_i(z)} \nabla f_i(z). \tag{16}$$

So we can instead solve the approximate problem

$$\begin{aligned} &\min_z f_0(z) + \mu \phi(z) \\ &\text{such that } Cz - b = 0. \end{aligned} \tag{17}$$

We define $z^*(\mu)$ to be the solution of (17) for some fixed $\mu > 0$. We denote the set of these solutions over all $\mu > 0$ to be denoted the *central path*, with each solution being referred to as a *central point*. The KKT conditions for the modified problem are simply

$$\begin{aligned} &\nabla f_0(z) + \sum_{i=1}^{n_\mathcal{I}} \left( -\frac{\mu}{f_i(z)} \right) \nabla f_i(z) + C^T \nu = 0 \\ &Cz - b = 0, \end{aligned} \tag{18}$$

for some $z \in \mathbb{R}^{n_z}$ and $\nu \in \mathbb{R}^{n_\mathcal{E}}$. Since the modified problem is still convex, finding variables $z$ and $\nu$ which satisfy (18) is sufficient for primal-dual optimality.

**Modified KKT Conditions**

Suppose some $z^*(\mu)$ solves (17) for some fixed $\mu > 0$. Further, suppose there exists $z$, $\lambda$, and $\nu$ which satisfy the *modified KKT conditions*

$$\begin{aligned} Cz - b &= 0 \\ f_i(z) &\le 0, \quad i \in [n_\mathcal{I}] \\ \lambda &\ge 0 \\ -\lambda_i f_i(z) &= \mu, \quad i \in [n_\mathcal{I}] \\ \nabla f_0(z) + \sum_{i=1}^{n_\mathcal{I}} \lambda_i \nabla f_i(z) + C^T \nu &= 0. \end{aligned} \tag{19}$$

The second, third, and fourth conditions imply that $\lambda_i > 0$ and $f_i(z) < 0$ for all $i \in [n_\mathcal{I}]$. So then we have $\lambda_i = \frac{\mu}{-f_i(z)}$, and so plugging in this formula for $\lambda_i$ into the last condition in (19) gets us that $z, \nu$

satisfy conditions (18). Because problem (17) is convex, this implies that $z$ is primal optimal and therefore $z = z^*(\mu)$. Hence, we get that satisfying modified KKT conditions (19) is sufficient for primal-dual optimality in problem (17).

Note how modified KKT conditions (19) only differ from ordinary KKT conditions by stating $-\lambda_i f_i(z) = \mu$ for $\mu > 0$ (instead of $\mu = 0$). This particular condition encourages variables to stay away from the boundaries imposed by inequality constraints. The larger $\mu$ is, the more our primal-dual iterates will stay away from those boundaries (hence the name *central* path).

## 1.3 Primal-Dual Interior-Point Methods

We will now cover a rudimentary primal-dual interior-point algorithm. Recall we're trying to solve a problem of the form (1) which may or may not be convex. We will add a slack variable so that the problem can now be expressed as

$$\begin{aligned} \min_{z,s}\ & f_0(z) \\ \text{such that}\ & f(z) + s = 0 \\ & h(z) = 0 \\ & s, \lambda \geq 0, \end{aligned} \tag{20}$$

where $f(z) := [f_1(z) \dots f_{n_\mathcal{I}}(z)]^T$ and $h(z) := [h_1(z) \dots h_{n_\mathcal{E}}(z)]^T$. The Lagrangian then takes the form

$$\mathcal{L}(z, s, \lambda, \nu) = f_0(z) + \lambda^T (f(z) + s) + \nu^T h(z). \tag{21}$$

Define the *residual* vector to be

$$r_\mu(y) = \begin{bmatrix} \nabla f_0(z) + \frac{\partial f}{\partial z}(z)^T \lambda + \frac{\partial h}{\partial z}(z)^T \nu \\ S\lambda - \mu \mathbf{1} \\ f(z) + s \\ h(z) \end{bmatrix} \tag{22}$$

where $y = (z, s, \lambda, \nu)$ and $S = \text{diag}(s)$ (and likewise $\Lambda = \text{diag}(\lambda)$). We also define $\mathbf{1}$ to be a vector of ones. Note how $r_\mu(y)$ encodes the modified KKT conditions. The purpose of a primal-dual interior-point algorithm then is to find $y$ such that $r_\mu(y) = 0$. In order to find such a $y$, our strategy will be to use Newton's method. From Taylor expansion we have that

$$r_\mu(y + p) \approx r_\mu(y) + \frac{\partial r_\mu}{\partial y}(y) p, \tag{23}$$

which we then set to zero and solve for the Newton step $p$. The resulting system is called the *KKT system*, which is

$$\begin{bmatrix} \nabla_z^2 \mathcal{L}(y) & 0 & \frac{\partial f}{\partial z}(z)^T & \frac{\partial h}{\partial z}(z)^T \\ 0 & \Lambda & S & 0 \\ \frac{\partial f}{\partial z}(z) & I & 0 & 0 \\ \frac{\partial h}{\partial z}(z) & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_z \\ p_s \\ p_\lambda \\ p_\nu \end{bmatrix} = - \begin{bmatrix} \nabla f_0(z) + \frac{\partial f}{\partial z}(z)^T \lambda + \frac{\partial h}{\partial z}(z)^T \nu \\ S\lambda - \mu \mathbf{1} \\ f(z) + s \\ h(z) \end{bmatrix}. \tag{24}$$

The left-hand side matrix (i.e., $\frac{\partial r_\mu}{\partial y}(y)$) is referred to as the *KKT matrix*.

We will now see the QP algorithm I have implemented for this project, which comes from [2]. The algorithm makes use of a technique called "Mehrotra's predictor-corrector", which essentially makes iterations more stable by keeping primal-dual iterates closer to the central path. It's somewhat out-of-scope for this report, but you can refer to [3] to learn more about it.

**At Each Iteration**

1. Evaluate the stopping criteria. In my project I simply used $\|r_\mu(y)\| < \epsilon$ for some small $\epsilon > 0$. Halt if criteria is satisfied.

2. Compute the affine scaling directions by solving

$$\begin{bmatrix} \nabla_z^2 \mathcal{L}(y) & 0 & \frac{\partial f}{\partial z}(z)^T & \frac{\partial h}{\partial z}(z)^T \\ 0 & \Lambda & S & 0 \\ \frac{\partial f}{\partial z}(z) & I & 0 & 0 \\ \frac{\partial h}{\partial z}(z) & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_z^{\text{aff}} \\ p_s^{\text{aff}} \\ p_\lambda^{\text{aff}} \\ p_\nu^{\text{aff}} \end{bmatrix} = - \begin{bmatrix} \nabla f_0(z) + \frac{\partial f}{\partial z}(z)^T \lambda + \frac{\partial h}{\partial z}(z)^T \nu \\ S\lambda \\ f(z) + s \\ h(z) \end{bmatrix}. \tag{25}$$

3. Compute both centering and corrector directions (added together) by solving

$$\begin{bmatrix} \nabla_z^2 \mathcal{L}(y) & 0 & \frac{\partial f}{\partial z}(z)^T & \frac{\partial h}{\partial z}(y)^T \\ 0 & \Lambda & S & 0 \\ \frac{\partial f}{\partial z}(z) & I & 0 & 0 \\ \frac{\partial h}{\partial z}(z) & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_z^{\text{cc}} \\ p_s^{\text{cc}} \\ p_\lambda^{\text{cc}} \\ p_\nu^{\text{cc}} \end{bmatrix} = - \begin{bmatrix} 0 \\ \sigma\mu\mathbf{1} - \text{diag}(p_s^{\text{aff}}) p_\lambda^{\text{aff}} \\ 0 \\ 0 \end{bmatrix}, \tag{26}$$

   where $\mu = \frac{s^T\lambda}{n_\lambda}$, $\sigma = \left( \frac{(s + \omega p_s^{\text{aff}})^T (\lambda + \omega p_\lambda^{\text{aff}})}{s^T \lambda} \right)^3$, and

$$\omega = \sup\{\omega \in [0,1] \mid s + \omega p_s^{\text{aff}} \geq 0,\ \lambda + \omega p_\lambda^{\text{aff}} \geq 0\}. \tag{27}$$

4. Update the primal and dual variables. First combine the two updates via

$$\begin{aligned} p_z &= p_z^{\text{aff}} + p_z^{\text{cc}} \\ p_s &= p_s^{\text{aff}} + p_s^{\text{cc}} \\ p_\lambda &= p_\lambda^{\text{aff}} + p_\lambda^{\text{cc}} \\ p_\nu &= p_\nu^{\text{aff}} + p_\nu^{\text{cc}}, \end{aligned} \tag{28}$$

   and set the step size so that $s$ and $\lambda$ remain nonnegative,

$$\alpha = \min\{1, 0.99 \sup\{\alpha \geq 0 \mid s + \alpha p_s \geq 0, \lambda + \alpha p_\lambda \geq 0\}\}. \tag{29}$$

5. Update the primal and dual variables:

$$\begin{aligned} z &:= z + \alpha p_z \\ s &:= s + \alpha p_s \\ \lambda &:= \lambda + \alpha p_\lambda \\ \nu &= \nu + \alpha p_\nu. \end{aligned} \tag{30}$$

6. Repeat from step 1.

# 2 Quadratic Programming

For my project, I used the primal-dual interior-point algorithm outlined in section 1.3 to solve a QP problem found in optimal control. In general, a QP problem has the form

$$\begin{aligned} &\text{minimize} \;\; z^T H z + g^T z \\ &\text{subject to} \;\; Pz \leq q, \; Cz = b. \end{aligned} \tag{31}$$

Using notation from previous sections, we have in this case $f_0(z) = z^T H z + g^T z$, $f(z) = Pz - q$, and $h(z) = Cz - b$. We will assume $H \succ 0$, $C$ is underdetermined, and that both $P$ and $C$ have full row rank. It is worth noting that because problem (31) is convex, any feasible $y$ at which $r_\mu(y) = 0$ must be primal-dual optimal. Thus, we don't have to worry about accidentally converging to a stationary point which isn't a global minimum. We will return to this issue when discussing NLP.

## 2.1 Modified KKT System for QP

In the QP case, the residual vector encoding the modified KKT conditions is

$$r_\mu(y) = \begin{bmatrix} 2Hz + g + P^T\lambda + C^T\nu \\ S\lambda - \mu\mathbf{1} \\ Pz - q + s \\ Cz - b \end{bmatrix}, \tag{32}$$

and its Jacobian with respect to $y$ is

$$\frac{\partial r_\mu(y)}{\partial y} = \begin{bmatrix} 2H & 0 & P^T & C^T \\ 0 & \Lambda & S & 0 \\ P & I & 0 & 0 \\ C & 0 & 0 & 0 \end{bmatrix}. \tag{33}$$

We then simply solve the QP problem using the algorithm outlined in section 1.3 using these matrices.

## 2.2 Optimal Control

Model Predictive Control (MPC) is a common method used for solving multiple-input multiple-output (MIMO) optimal control problems. Suppose we have a dynamical system with state $x_t$ at time step $t$, and that $x$ evolves over discrete time steps according to a linear time-invariant system $x_{t+1} = Ax_t$. Control is used in cases when we can input "energy" $u$ into the system, in order to guide the state $x$ to a target called a *setpoint*. In this scenario, the linear time-invariant system will take the form

$$x_{t+1} = Ax_t + Bu_t. \tag{34}$$

Suppose we are sitting at state $x_0$ at time step $t = 0$, and we want to optimize over a trajectory $\tau = (u_0, x_1, \ldots, u_{T-1}, x_T)$ which maximizes some objective function ($T$ in this case is called the *prediction horizon*). We would like to define an objective function which penalizes the state and input vectors for straying from their setpoints. Further, in many real-world scenarios, it isn't feasible to allow arbitrarily large difference in inputs between consecutive time steps; for example, no SUV can go from 0 to 60 MPH in under a second. Thus, we would like to penalize these differences as well. We define our objective function to be

$$\sum_{t=0}^{T-1} \left( x_t^T Q x_t + q^T x_t + u_t^T R u_t + r^T u_t + \Delta u_t^T S \Delta u_t \right) + x_T^T Q_T x_T + q_T^T x_T \tag{35}$$

for $Q, R, S, Q_T \succ 0$ and $\Delta u_t = u_t - u_{t-1}$. Note that it is customary for MPC methods to impose a terminal cost on $x_T$ for reasons beyond the scope of this report (hence why we're allowing $Q_T$ and $q_T$ to differ from $Q$ and $q$). Furthermore, we can also apply inequality constraints to the states, inputs, and input differences as well. These constraints will be linear and take the form

$$F_x x_t \leq f_x, \quad F_u u_t \leq f_u, \quad F_{\Delta u} \Delta u_t \leq f_{\Delta u}, \quad F_T x_T \leq f_T. \tag{36}$$

To briefly summarize our control problem, we would like to choose an optimal trajectory $\tau = (u_0, x_1, \ldots, u_{T-1}, x_T)$ which minimizes objective (35) subject to inequality constraints (36) and equality constraints (34). All that's left for us to do is to formulate this as a QP problem like (31). To make it easier to read, we will derive parameters for the $T = 3$ case. First let's begin with the objective function, which can be expressed as $z^T H z + g^T z$ where

$$H = \begin{bmatrix} R+2S & 0 & -S & 0 & 0 & 0 \\ 0 & Q & 0 & 0 & 0 & 0 \\ -S & 0 & R+2S & 0 & -S & 0 \\ 0 & 0 & 0 & Q & 0 & 0 \\ 0 & 0 & -S & 0 & R+S & 0 \\ 0 & 0 & 0 & 0 & 0 & Q_T \end{bmatrix}, \quad g = \begin{bmatrix} -2Su_{-1} + r \\ q \\ r \\ q \\ r \\ q_T \end{bmatrix}, \quad \text{and} \quad z = \begin{bmatrix} u_0 \\ x_1 \\ u_1 \\ x_2 \\ u_2 \\ x_3 \end{bmatrix}. \tag{37}$$

Note how $x_0$ isn't a part of $z$ since it's fixed (and hence not a decision variable). Since we're penalizing $\Delta u_t$, we must include another fixed variable $u_{-1}$ which represents the input in the last time step prior to arriving at state $x_0$. For the inequality constraints, we define

$$P = \left[\begin{array}{cccccc} F_u & 0 & 0 & 0 & 0 & 0 \\ F_{\Delta u} & 0 & 0 & 0 & 0 & 0 \\ 0 & F_x & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & F_u & 0 & 0 & 0 \\ -F_{\Delta u} & 0 & F_{\Delta u} & 0 & 0 & 0 \\ 0 & 0 & 0 & F_x & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & F_u & 0 \\ 0 & 0 & -F_{\Delta u} & 0 & F_{\Delta u} & 0 \\ 0 & 0 & 0 & 0 & 0 & F_T \end{array}\right], \quad \text{and} \quad q = \left[\begin{array}{c} f_u \\ f_{\Delta u} + F_{\Delta u} u_{-1} \\ f_x \\ \hline f_u \\ f_{\Delta u} \\ f_x \\ \hline f_u \\ f_{\Delta u} \\ f_T \end{array}\right]. \tag{38}$$

Note that I included horizontal lines in (38) just to make the pattern more discernible. Lastly, for the equality constraints we define

$$C = \begin{bmatrix} -B & I & 0 & 0 & 0 & 0 \\ 0 & -A & -B & I & 0 & 0 \\ 0 & 0 & 0 & -A & -B & I \end{bmatrix}, \quad \text{and} \quad b = \begin{bmatrix} Ax_0 \\ 0 \\ 0 \end{bmatrix}. \tag{39}$$

For different values of $T$, simply expand or shrink the matrices above using the same pattern. We have now fully specified our control problem as a QP of the form (31). The next section will go through some of the results I've obtained using this setup.

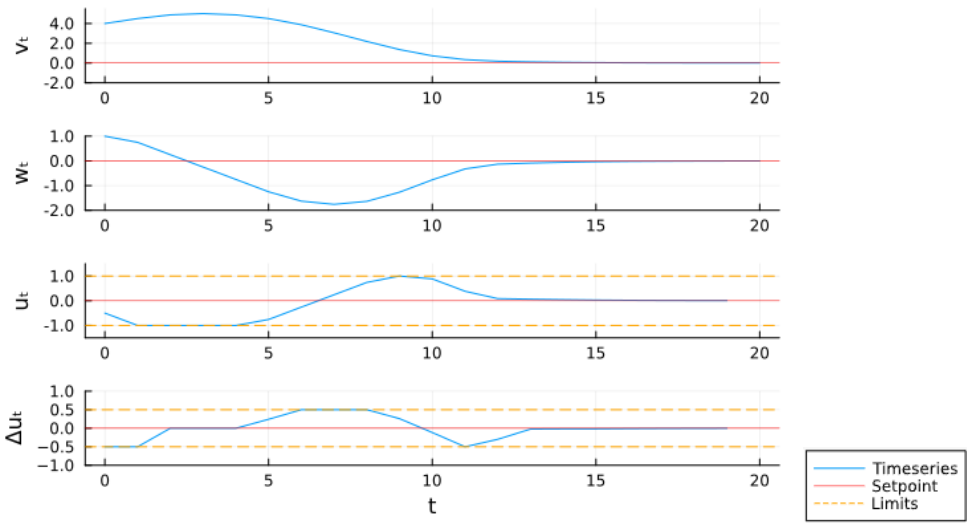


Figure 1: The optimal trajectory for the double integrator across discrete time steps $t = 1, \ldots, 20$. See how variables obey their constraint limits and converge to their setpoints.

### 2.3 Example & Results

In this section, we will apply what we have learned to an example control problem. The double integrator is an LTI system with dynamics

$$\begin{bmatrix} v_{t+1} \\ w_{t+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_t \\ w_t \end{bmatrix} + \begin{bmatrix} 0 \\ \Delta t \end{bmatrix} u_t, \tag{40}$$

with state $x_t = [v_t, w_t]^T$, scalar input $u_t$, and frequency parameter $\Delta t$. For this experiment, we set $\Delta t = 0.5$. Our goal is to drive the state to a setpoint $d \in \mathbb{R}^2$. For this example, we will simply drive the state vector to the origin by setting $d = 0$.

We now must define the objective function. We set $Q = I$ and $q = 0$. This way, the objective penalizes proportional to how far away the state is from the origin. Further, we shall set $R = S = 0.01$ and $r = 0$, which simply penalizes the input proportional to how large it is in absolute value. We also set the initial state to be $x_0 = [4 \;\; 1]^T$, and the prediction horizon to $T = 20$.

Lastly, we impose the following constraints on the states, inputs, and input differences at each time step:

$$\begin{bmatrix} -5 \\ -2 \end{bmatrix} \leq \begin{bmatrix} v_t \\ w_t \end{bmatrix} \leq \begin{bmatrix} 5 \\ 2 \end{bmatrix}, \quad -1 \leq u_t \leq 1, \quad \text{and} \quad -0.5 \leq \Delta u_t \leq 0.5. \tag{41}$$

The optimal trajectory output by my QP solver is displayed as Figure 1.

## 3 Nonlinear Programming

In this section, we won't look at the most general form of nonlinear programming, but rather, we will look at a formulation which will allow us to tackle an optimal control problem similar to the one covered in section 2, but this time with nonlinear dynamics. Recall that the dynamics is encoded into a QP problem through

the affine equality constraints $Cz = b$. Under nonlinear dynamics however, these equality constraints are not affine. The problem we are now trying to solve is

$$\begin{aligned} &\text{minimize} \quad z^T H z + g^T z \\ &\text{subject to} \quad Pz \leq q, \; h(z) = 0 \end{aligned} \tag{42}$$

where $h$ isn't affine. The residual vector encoding the modified KKT conditions is

$$r_\mu(y) = \begin{bmatrix} 2Hz + g + P^T\lambda + \frac{\partial h}{\partial z}(z)^T\nu \\ S\lambda - \mu\mathbf{1} \\ Pz - q + s \\ h(z) \end{bmatrix}, \tag{43}$$

and its Jacobian with respect to $y$ is

$$\frac{\partial r_\mu(y)}{\partial y} = \begin{bmatrix} \nabla_z^2\mathcal{L}(y) & 0 & P^T & \frac{\partial h}{\partial z}(z)^T \\ 0 & \Lambda & S & 0 \\ P & I & 0 & 0 \\ \frac{\partial h}{\partial z}(z) & 0 & 0 & 0 \end{bmatrix}. \tag{44}$$

Equation (44) contains $\nabla_z^2\mathcal{L}(y)$ which in general requires a tensor for the second-order derivative of $h(z)$. One can compute this Hessian explicitly, but I decided this would be a good opportunity for me to learn about quasi-Newton methods as a convenient way for me to bypass the calculation.

Problem (42) is nonconvex, meaning that there are possibly several stationary points $y$ which aren't primal-dual optimal, but for which $r_\mu(y) = 0$; we will discuss how to ensure we at least converge to a local minimum.

## 3.1 Ensuring Descent

Since the algorithm outlined in section 1.3 only seeks to locate KKT points, we may very well end up converging to a maximum or another type of stationary point. This section will outline some basic techniques outlined in [4] which we can use to improve the robustness of our algorithm.

Recall that our ultimate goal is to minimize $f_0(z)$ subject to constraints. When we take a Newton step at each iteration, we would like to ensure that a given step addresses the two (conflicting) goals of minimizing $f_0$, while also bringing the primal-dual iterates closer to satisfying their constraints. One way to do this is by a running line search on an objective function which rewards us for minimizing $f_0$ and punishes us for not satisfying constraints. We call this objective function a *merit function*. One simple formulation of a merit function is

$$\kappa_\rho(z, s) = f_0(z) - \mu \sum_{i=1}^{n_\lambda} \log s_i + \rho\|f(z) + s\|_2 + \rho\|h(z)\|_2, \tag{45}$$

with $\rho > 0$. One concern may be that the Newton step $p$ obtained by solving the KKT system $\frac{\partial r_\mu}{\partial y}(y)p = -r_\mu(y)$ may not be a descent direction for merit function (45). In this case, no step size small enough would yield a sufficient decrease in $\kappa_\rho$, causing line search to fail. The following theorem will establish when $p$ is a descent direction for a broad class of merit functions, which includes (45).

**Theorem 3.1.** *Suppose we obtain a Newton step $p$ from the modified KKT system (24). This KKT system is equivalent to*

$$\begin{bmatrix} \nabla_z^2 \mathcal{L}(y) & 0 & \frac{\partial f}{\partial z}(z)^T & \frac{\partial h}{\partial z}(z)^T \\ 0 & S^{-1}\Lambda & I & 0 \\ \frac{\partial f}{\partial z}(z) & I & 0 & 0 \\ \frac{\partial h}{\partial z}(z) & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_z \\ p_s \\ p_\lambda \\ p_\nu \end{bmatrix} = - \begin{bmatrix} \nabla f_0(z) + \frac{\partial f}{\partial z}(z)^T \lambda + \frac{\partial h}{\partial z}(z)^T \nu \\ \lambda - \mu S^{-1} \mathbf{1} \\ f(z) + s \\ h(z) \end{bmatrix}, \tag{46}$$

*which is obtained by left-multiplying the second block row by $S^{-1}$. We find that the KKT matrix on the left-hand side is now symmetric. Refer to this matrix as $K$. We have that $p = \begin{bmatrix} p_z \\ p_s \end{bmatrix}$, is a descent direction for merit function (45) if the inertia of $K$ is $(n_z + n_\lambda,\ n_\nu + n_\lambda,\ 0)$.*

There are cases in which the rank of KKT matrix $K$ is not in fact $(n_z + n_\lambda,\ n_\nu + n_\lambda,\ 0)$; namely, this happens when $\nabla_z^2 \mathcal{L}(y)$ loses a positive eigenvalue (which results in $K$ having less positive eigenvalues than it should), or when $\frac{\partial h}{\partial z}(z)$ loses rank (which results in $K$ having more positive eigenvalues than it should). When this occurs, the KKT matrix can be modified to be

$$\begin{bmatrix} \nabla_z^2 \mathcal{L}(y) + \delta I & 0 & \frac{\partial f}{\partial z}(z)^T & \frac{\partial h}{\partial z}(z)^T \\ 0 & S^{-1}\Lambda & I & 0 \\ \frac{\partial f}{\partial z}(z) & I & 0 & 0 \\ \frac{\partial h}{\partial z}(z) & 0 & 0 & -\gamma I \end{bmatrix}, \tag{47}$$

where at each iteration a small $\delta > 0$ and $\gamma > 0$ are found which maintain the inertia of $K$ at the appropriate value. Note that we don't have to worry about $\frac{\partial f}{\partial z}(z)$ since $[\frac{\partial f}{\partial z}(z) \ \ I]$ always has full row rank. More about this can be in read in [4].

In this report, we will estimate $\nabla_z^2 \mathcal{L}(y)$ using damped BFGS updates. The BFGS method not only prevents us from having to compute a complicated Hessian matrix, but since we're using its *damped* version, we are guaranteed to have an estimate of $\nabla_z^2 \mathcal{L}(y)$ that's positive definite. In standard BFGS, in order to maintain positive definiteness one has to ensure that one's step sizes always satisfies the strong Wolfe conditions. However, I personally found this caveat to complicate things, so I opted for the damped BFGS updates instead.

In the next section, we will find that the Jacobian of our equality constraint vector $h$ always maintains full row rank. Hence, the modifications to the KKT matrix shown in (47) aren't necessary, as our KKT matrix will always maintain the correct balance of positive and negative eigenvalues.

## 3.2 Nonlinear Optimal Control

Recall from section 2.2 that a linear time-invariant system has the form

$$x_{t+1} = Ax_t + Bu_t. \tag{48}$$

This section however will be about *nonlinear* time-invariant systems of the form

$$x_{t+1} = f(x_t, u_t), \tag{49}$$

where $f$ isn't affine. In the case of linear dynamics, the dynamics is encoded into a QP problem using the affine equality constraints $Cz = b$. Because our dynamics are no longer linear, the equality constraints are no longer affine. Hence, we solve the optimization problem (42), for a non-affine $h(z)$.

For our optimal control problem, recall $z = (u_0, x_1, \ldots, u_{T-1}, x_T)$. We have that

$$h(z) = \begin{bmatrix} x_1 - f(x_0, u_0) \\ x_2 - f(x_1, u_1) \\ \vdots \\ x_T - f(x_{T-1}, u_{T-1}) \end{bmatrix}, \tag{50}$$

and its Jacobian (in the easier case of $T = 3$) is equal to

$$\frac{\partial h}{\partial z}(z) = \begin{bmatrix} -f_u(x_0, u_0) & I & 0 & 0 & 0 & 0 \\ 0 & -f_x(x_1, u_1) & -f_u(x_1, u_1) & I & 0 & 0 \\ 0 & 0 & 0 & -f_x(x_2, u_2) & -f_u(x_2, u_2) & I \end{bmatrix}. \tag{51}$$

Note how the above matrix has full row rank (as mentioned in the previous section). We now have all the tools necessary to numerically solve this optimal control problem under nonlinear system dynamics.

### 3.3 Example & Results

For this second example, we will attempt to solve a nonlinear control problem. The pendulum equation describes the nonlinear dynamics of a pendulum holding a point mass, and is given by

$$\ddot{\theta} = -\frac{g}{l}\sin\theta - \frac{k}{m}\dot{\theta} + \frac{1}{ml^2}u, \tag{52}$$

where $\theta$ is the angle of the pendulum from its downwards position going counter-clockwise, $m$ is the mass of the point mass, $l$ is the length of the pendulum, $k$ is the drag coefficient if we were to assume friction, and $g$ is gravity. Further, we have that $u$ is torque input applied to the point mass (going counter-clockwise). For this example, we set $m = l = k = 1$, and $g = 9.8$. We are now left with the equation

$$\ddot{\theta} + \dot{\theta} + g\sin\theta = u. \tag{53}$$

We would like to turn this equation into a first-order differential equation. To do so, set $x_1 = \theta$, and $x_2 = \dot{\theta}$. We can then define $f(x, u)$ so that

$$\dot{x} = f(x, u) := \begin{bmatrix} x_2 \\ -g\sin x_1 - x_2 + u \end{bmatrix}. \tag{54}$$

Continuous-time LTI systems of the form $\dot{x} = Ax + Bu$ can be converted into its discrete-time equivalent $x_{k+1} = Ax_k + Bu_k$ exactly. However, the dynamics above isn't linear (note the presence of $\sin x_1$). Therefore, we will have to approximate the discrete-time dynamics using Euler's method. We have that

$$x_{k+1} \approx x_k + f(x_k, u_k)\Delta t. \tag{55}$$

We assume our discrete-time dynamics to then be given by $g(x, u) := x + f(x, u)\Delta t$, and so $x_{k+1} = g(x_k, u_k)$. Because our algorithm also requires the Jacobian of the equality constraints, we have

$$\frac{\partial g}{\partial x}(x, u) = I + \frac{\partial f}{\partial x}(x, u)\Delta t, \quad \text{and} \quad \frac{\partial g}{\partial u}(x, u) = \frac{\partial f}{\partial u}(x, u)\Delta t. \tag{56}$$

Our goal within this experiment is to balance the pendulum upright in a stable fashion. In other words, we would like to push the state vector towards the setpoint $d = \begin{bmatrix} \pi \\ o \end{bmatrix}$.

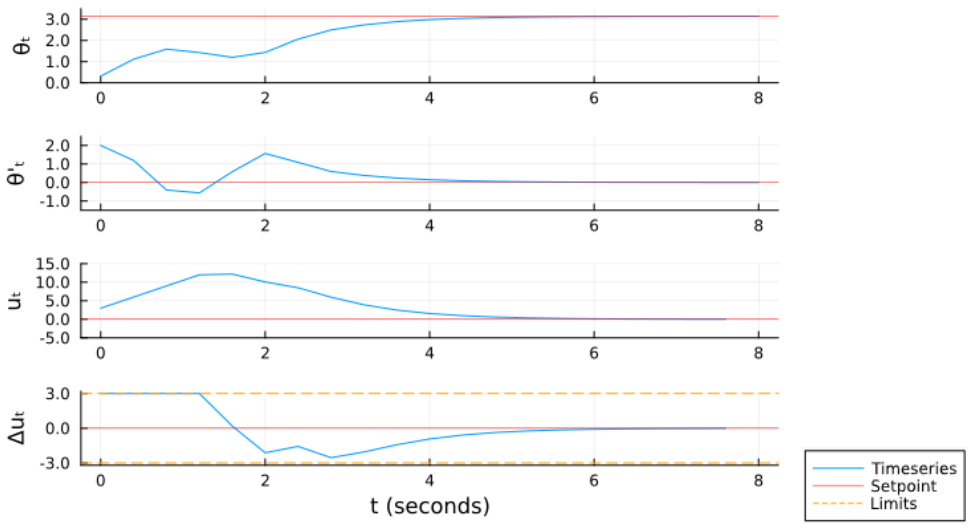


Figure 2: An optimal trajectory of the pendulum system output by my controller using $T = 20$ discrete time steps of interval $\Delta t = 0.4$ seconds. Note that this trajectory is approximated using Euler's method.

For the objective function parameters, we set $Q = Q_T = I$ and $q = -2d$, as we would like to minimize the distance between $x$ and $d$, the square of which is $(x-d)^T(x-d) = x^T I x - 2d^T x + d^T d$ (we can ignore the constant $d^T d$). We also set $R = S = 0.01$, and $r = 0$ as we simply want to penalize the excessive use of torque, and large jumps in torque input between time steps. Lastly, we impose the constraints

$$-2 \leq \Delta u_t \leq 2 \tag{57}$$

for each time step $t$. The optimal trajectory output by my NLP solver is graphed in Figure 2.

I should mention that the solver I implemented for this example can be somewhat finicky, especially when imposing really strict inequality constraints, or making the dynamics more nonlinear by making $g$ large. Furthermore, in order to save time I simply set the $\rho$ parameter in (45) to be a very large number, but ideally it would be better to iteratively update this parameter as the algorithm converges.

# 4 Conclusion

This report provided a concise summary for many of the fundamental topics needed to understand primal-dual interior-point methods for convex and nonconvex problems. Interesting applications from the field of control theory were included to give readers a sense of how these methods can be put into practice. Furthermore, readers got to see how nonlinearity in the context of optimal control leads to nonconvexity in the context of optimization. One last thing that readers should note is that this project was created for pedagogical purposes, so much of the code I implemented isn't optimized for performance (although it still runs sufficiently fast for the purposes of this project).

# References


[1] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[2] Jacob Mattingley and Stephen Boyd. Cvxgen: A code generator for embedded convex optimization. *Optimization and Engineering*, 13:1–27, 2012.

[3] Sanjay Mehrotra. On the implementation of a primal-dual interior point method. *SIAM Journal on optimization*, 2(4):575–601, 1992.

[4] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.

[5] Yang Wang and Stephen Boyd. Fast model predictive control using online optimization. *IEEE Transactions on control systems technology*, 18(2):267–278, 2009.