# Handling Inverse Optimal Control Problems using Evolutionary Bilevel Optimization*

Varun Suryan
Department of Mechanical Engineering
Indian Institute of Technology Jodhpur,India
Email: ug201212038@iitj.ac.in

Ankur Sinha
Production and Quantitative Methods
Indian Institute of Management Ahmedabad,India
Email: asinha@iimahd.ernet.in

Pekka Malo
Department of Information and Service Economy
Aalto University School of Business,Finland
Email: pekka.malo@aalto.fi

Kalyanmoy Deb
Computer Science and Engineering
Michigan State University,USA
Email: kdeb@egr.msu.edu

**COIN Report: 2016017**

**Abstract**

Optimal Control is a task where it is desired to determine the inputs of a dynamical system that optimize (minimize or maximize) a specified cost functional also known as performance index while satisfying any constraints on behaviour of the system. As the name suggests, Inverse Optimal Control is the opposite of former one and thus is associated with mining of the cost functional, optimal behaviour of which fits the given results best. In this paper, we present the importance of evolutionary bilevel optimization techniques as a promising approach to solve inverse optimal control problems. Generally inverse optimal control problems are found to be ill posed which makes them computationally expensive in addition to associated redundancy with solution. Inverse optimal control theory works as a stepping stone in figuring out the underlying optimality criteria in a given task. It has several other applications in areas like Markov's Decision Processes and Game Theory. In our work, we solve inverse optimal control

*Accepted for presentation at WCCI-2016 (25-29 July 2016), Vancouver, Canada.

problems to retrieve the original functional in optimal control task using metaheuristic based bilevel optimzation techniques. The dataset comprising of state variables generated from an optimal control problem with an unknown functional is utilized to mine the functional. In the later part of paper, we formulate a problem of human motion transfer as an bilevel optimization task, subsequently run our algorithm on it and benchmark our results.

# 1 Introduction

Optimal control theory is a branch of mathematical optimization that has enormous applications in a wide variety of areas in science and engineering. Optimal control problems involve calculation of state functions along with control sequences which minimize a given cost functional often known as performance index subject to some constraints. There are two kinds of techniques named as Direct and Indirect methods to tackle any general optimal control problem. Indirect methods are analytical techniques that involve the principles from variational calculus derived from the early works of Pontryagin [1] and Bellman [2]. Direct methods deploy the use of numerical computations and approximations for calculation of optimal control law and corresponding state funcions[3, 4]. One of the major challenges in control theory is deriving the performane index or reward function which fits best on the given data set or demonstrations. Such kind of tasks lie in the category of Inverse Optimal Control theory which is quite analogous to inverse optimzation problems. Inverse problems solicit the calculation of cause based on the given result. It means that such tasks start with result and go to calculation of causes which is quite contrary to general problems we come across in science. Inverse problems have wide application spanning optics, radar, acoustics, communication theory, signal processing, medical imaging, computer vision, geophysics, oceanography, astronomy, remote sensing, natural language processing, machine learning, nondestructive testing, and many other fields. We present a novel approach that uses evolutionary bilevel optimization [refer section 2] principles, which could be useful in solving complex inverse optimal control problems. In a bilevel formulation of inverse optimal control problem we try to minimize the error between the *experimental data* and *computed data* at upper level, while the lower level optimization task involves solving an optimal control problem for a sample functional. Solving the lower level problem iteratively with different functionals allows to search for a functional that deviates the least from the experimental data.

Rest of the paper is organized as follows. In the next section, we introduce general bilevel optimization problems and their solution techniques. Thereafter, we provide a comprehensive study on inverse optimal control problems along with discussion of research on its solution methodologies which is followed by discussion on formulating such problems as bilevel optimization problems. Lastly, we present numerical results on two types of inverse optimal control problems. The problems have been chosen such that analytical solutions can be computed in order to test the performance and applicability of evolutionary bilevel optimization techniques to inverse optimal control problems.

# 2 Bilevel Optimization

Bilevel optimization is a two level optimization task where one task is nested in another optimization task. The outer task generally known as upper level optimization task and the inner one is called lower level optimization task. Accordingly, in such problems we need to tackle with two kinds of variables which are called upper level variables and lower level variables often known as upper level decision vectors and lower level decision vectors respectively. A general bilevel optimzation problem is defined as follows:

**Definition.** *For the upper-level objective function* $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ *and lower-level objective function* $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$

$$\begin{aligned}
\underset{x_u \in X_U, x_l \in X_L}{\text{minimize}} \quad & F(x_u, x_l) \\
\text{subject to} \quad & x_l \in \underset{x_l}{\text{argmin}}\{f(x_u, x_l) : \\
& \qquad g_j(x_u, x_l) \leq 0, j = 1, \ldots, J \\
& \qquad h_k(x_u, x_l) = 0, k = 1, \ldots, K\} \\
& G_l(x_u, x_l) \leq 0, l = 1, \ldots, L \\
& H_m(x_u, x_l) = 0, m = 1, \ldots, M
\end{aligned}$$

As is clear from the above definition that $x_u, x_l$ are upper level and lower level decision vectors respectively. $G_l, g_j$ are upper level and lower level inequality constraints respectively. Equality constraints at upper level and lower level are defined as $H_m$ and $h_k$ respectively.

Bilevel optimization tasks were first realized by German economist Stackelberg who described these problems as hierarchical tasks in his popular work on leader-follower games also referred as Stackelberg games [5]. In Stackelberg games, players compete with each other and leader makes the first move knowing the ex ante that follower knows about its actions. Hence, leader's task is to optimize its response keeping in the mind that follower also wants to optimize theirs which results in a nested optimization in which lower level task corresponds to follower's optimization problem and upper level task is taken care by leader. Bilevel optimization has further applications in toll-setting problems [6–9], environmental economics [10–12], chemical industry [13–15], optimal design [16–18] and defense [19–22], to name a few.

Bilevel optimization problems are classified as strongly NP-hard task. Only an optimal solution to lower level task makes a possible feasible candidate to the upper level task. This requirement makes these problems computationally very expensive since we need to solve lower level task every time for each iteration at the upper level. A number of strategies to tackle such problems have been discovered and discussed by mathematicians. Some of the popular techniques to solve such problems are classical approaches like Karush-Kuhn-Tucker (KKT) that reduces a bilevel optimization task to a single-level optimization problem. For instance, in case of linear bilevel problems using KKT conditions transforms the problem to mixed integer linear program. Approaches based on vertex enumeration [23–25], as well as branch-and-bound [26, 27] have been proposed to solve linear bilevel problems. Branch-and-bound approach has also been to handle single-level reductions of linear-quadratic [28] and quadratic-quadratic [29, 30] bilevel problems. Other classical optimization techniques include descent-based approaches [31, 32], use of penalty functions [?, 33, 34] and trust-region methods [35–37]. As soon as complexity of the bilevel optimization increases, the classical bilevel techniques cease to substantiate their utility and we need to rely on computationally heavy techniques like evolutionary algorithms to solve the problem.

## 2.1 Evolutionary Bilevel Optimization

Most of the earlier evolutionary algorithms for solving bilevel problems are nested in structure, where lower level task is executed for every upper level decision vector as illustrated in figure 1. Mathieu's work [38] towards using genetic algorithms for bilevel linear programming was one of the

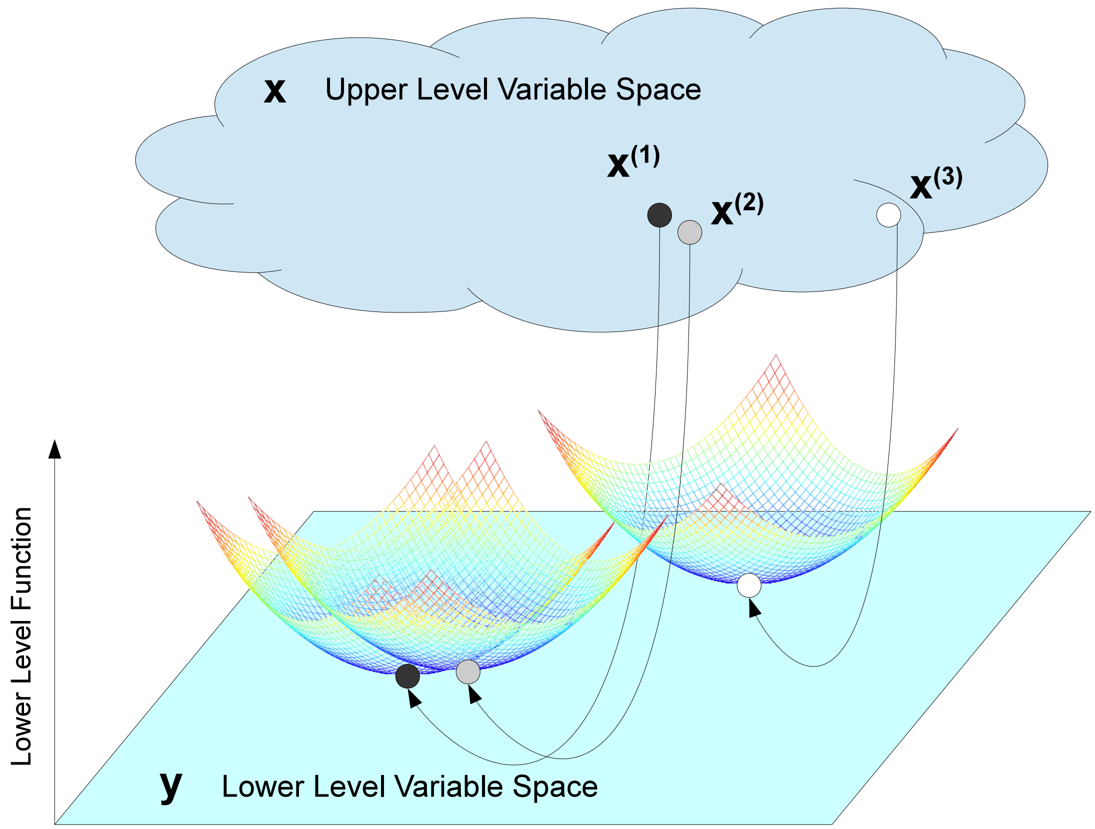


Figure 1: Lower level optimal solutions for various different upper level members.

first studies where a hierarchical programming task was solved using an evolutionary techniques. In the original work [38], the upper level task was solved using genetic algorithms (GAs), while a classical approach was used at the lower level. In another work, Yin [39] handled the lower level using the Frank-Wolfe algorithm and deployed genetic algorithm at upper level task for solving non-convex bilevel optimization problems. Wang et al. [40, 41] presented another solution methodology to tackle non-linear bilevel programming problems by applying Karush-Kuhn-Tucker optimality conditions at the lower level and further applying heuristic methods. More recently, metamodeling-based solution methods have been used for solving bilevel optimization problems. Some of the recent algorithms that rely on the reaction set mapping are [42–44]. In this paper, we will use the Bilevel Evolutionary Algorithm based on Quadratic Approximations (BLEAQ) proposed in [42, 43] to handle the inverse optimal control problems.

# 3 Inverse Optimal Control Theory

Optimal control is a mathematical field which is concerned with control policies that can be deduced using optimization algorithms on cost functional. In Mathematics, functional is a name given to entities which are a function of functions. These state and control parameters are further a function of some independent parameter. Feng Lin discusses the various optimal control problems formulation and their implications in various daily routines like Minimal Tracking Error Problems, Minimal Energy Problem and Linear Quadratic Regulator (LQR) Problem in fourth chapter of his famous book in control theory [45]. A general optimal control problem is being formulated as follows:

**Definition.** *Minimize the continuous time cost functional*

$$J = \Phi[\boldsymbol{x}(t_0), t_0, \boldsymbol{x}(t_f), f] + \int_{t_0}^{t_f} \mathcal{L}[\boldsymbol{x}(t), \boldsymbol{u}(t), t] dt$$

*subject to first order dynamic constraints*

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{a}[\boldsymbol{x}(t), \boldsymbol{u}(t), t],$$

*the algebraic path constraints*

$$\boldsymbol{b}[\boldsymbol{x}(t), \boldsymbol{u}(t), t] \leq \boldsymbol{0},$$

*and the boundary conditions*

$$\Psi[\boldsymbol{x}(t_0), t_0, \boldsymbol{x}(t_f), t_f] = 0$$

where $\mathbf{x}$(t) is the state, $\mathbf{u}$(t) is the control, $t$ is the independent variable (generally speaking, time), $t_0$ is the initial time, and $t_f$ is the terminal time. The objective here is to find state function $\mathbf{x}(t)$ and control sequence $\mathbf{u}(t)$ which optimize the cost functional $J$.
Several solution techniques for general optimal control problems which have been discussed in literature comprise of indirect methods such as Hamilton-Jacobi-Bellman Equation and direct methods whih rely on numerical computations of control and input policies by the help of boundary value problems.

## 3.1 Recent research on inverse optimal control theory

As already discussed in section 1, Inverse problems have various applications in different fields. Inverse optimality also known as Inverse Reinforcement Learning (IRL) has attracted considerable attention in both control engineering and machine learning. Inverse optimality was first studied for control-theoretic purposes in relation to stability[46, 47]. The idea was further extended by Krstic et al.[48] where one designs a control-Lyapunov function, treats it as an optimal value function (and derives the corresponding control law) and finds the cost for which this value function is optimal. Ziebart et al.[49] propsed the maximum entropy model to tackle inverse optimal control problems in 2008, in which they solve forward problem repeatedly in the inner loop. Most of the past methods for IRL use linear combinations of functions in forward problem due to the computational complexity involved in finding the solution which restrict the search space, albeit we have also used linear combination to demonstrate the applicability of evolutionary bilevel algorithms to such problems. However, evolutionary bilevel algorithms can be easily applied to more complex non-linear combinations without additional modifications.

## 3.2 Formulation of Inverse Optimal Control problem as Bilevel Optimization Problem

An inverse optimal control problem consists of determining the objective function $\mathcal{L}(\mathbf{x}(t), \mathbf{u}(t), t)$ that produces the best fit to the dataset in the least squares sense. For the objective function we make the assumption that it can be expressed as a weighted sum of a series of $n$ base functions

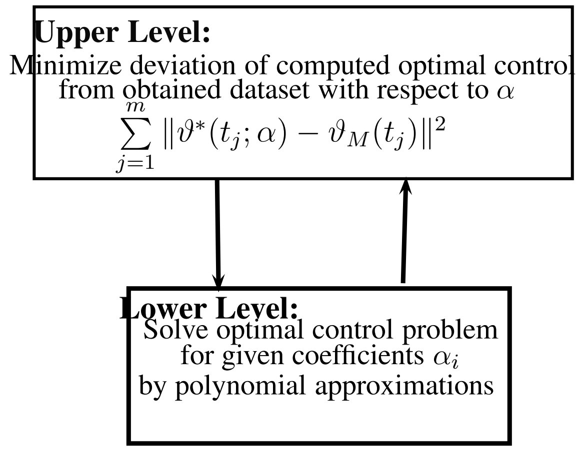


Figure 2: Inverse optimal control problem formulated as bilevel optimization problem. Upper level passes $\alpha$ parameters to lower level and lower level passes optimal control $\vartheta^*$ for respective alpha values to upper level.

$\phi_i(t)$ with corresponding weight parameters $\alpha_i$ :

$$\mathcal{L}(\mathbf{x}(t), \mathbf{u}(t), t) = \sum_{i=0}^{n-1} \alpha_i \phi_i(\mathbf{x}(t), \mathbf{u}(t), t)$$

Thus, the problem of obtaining the best objective function $\mathcal{L}(\cdot)$ is transformed into the getting the best weight factors $\alpha_i$[50] since $\mathcal{L}(\cdot)$ is being expressed as linear combination of respective base functions. Now, the weight factors play the crucial role in determining the objective function because greater the weight, more the respective term is punished. On the other extremity, $\alpha_i$ may vanish for some term. A simplified version of inverse optimal control as a bilevel optimization task can be formulated with upper level and lower level as illustrated in figure 2 :

$$\begin{aligned}
&\underset{\alpha}{\text{minimize}} \quad \sum_{j=1}^{m} \|\vartheta^*(t_j;\alpha) - \vartheta_M(t_j)\|^2 \\
&\qquad \text{where } \vartheta^*(t_j;\alpha) \text{ is a solution of,} \\
&\underset{x,u,T}{\text{minimize}} \int_0^T \left( \sum_{i=1}^{n} \alpha_i \phi_i(\mathbf{x}(t), \mathbf{u}(t), t) \right) dt \\
&\qquad \text{subject to} \\
&\qquad\quad \dot{\mathbf{x}} = \mathbf{a}[\mathbf{x}(t), \mathbf{u}(t), t] \\
&\qquad\quad \mathbf{x}(0) = \mathbf{x}_0, \mathbf{x}(T) = \mathbf{x}_e
\end{aligned}$$

where $\mathbf{x}$ summarizes the full or reduced vector of states and controls, $\vartheta(t)^T = (\mathbf{x}(t)^T, \mathbf{u}(t)^T)$ and $\vartheta_M$ denotes the measured values or dataset.

# 4 Numerical\Direct Solution to Inverse Optimal Control Problems

Inverse problems are quite expensive to solve numerically in comparison to forward problems since they are not often well-posed. A problem is to be called well-posed if it satisfies following three properties.

1. there is a solution,
2. the solution is unique, and
3. the solution depends continuously on the data.

Inverse problems are found to be ill-posed generally. In this section we present the numerical solution of inverse optimal control problems treated as bilevel problem. The upper level handles the iteration over $\alpha$ such that the fit between measurements and optimal control problem solution (which is being calculated at lower level) is improved. Each upper level iteration includes one call to the lower level where a forward optimal control problem is solved for the current set of $\alpha_i$ .The optimal solution of this problem is then communicated back to the upper level such that the least squares fit between measurements and computations can be evaluated for this iteration.Optimal control problem at the lower level can be solved analytically using principles of Pontryagin but complexity issues and accuracy requirements force us to work out numerical methods so we have adopted the numerical approach using polynomial approximations [51,52]. In our work, a five degree polynomial has been used to approximate the state functions and control sequences which optimize the given performance index at lower level.

Now we present our result and calculations on solving inverse optimal control problems formulated as bilevel task using our algorithm by running it on several example problems.

## 4.1 Result

### 4.1.1 Example 1

The first example problem depicts a linear system which is the modified version of optimal control problem described in [51] with $x_1(t), x_2(t)$ as state parameters and $u(t)$ as control parameter.

Find $u^*(t)$ that minimizes

$$J = \int_0^1 [\alpha_1 x_1^2(t) + \alpha_2 x_2^2(t) + \alpha_3 u^2(t)]dt \tag{1}$$

subject to

$$\dot{x}_1 = x_2,$$
$$\dot{x}_2 = -x_2 + u,$$

with initial conditions

$$x_1(0) = 0, x_2(0) = -1$$

and

$$(\alpha_1, \alpha_2, \alpha_3) = (15, 10, 0.1)$$

The first step in solving such problems by proposed method is approximating $x_1(t)$ by $5^{th}$ degree polynomial of unknown parameters, we assume

$x_1(t) = t^5 + c_4t^4 + c_3t^3 + c_2t^2 + c_1t + c_0$

Then, $\dot{x}_1(t)$ is calculated which subsequently gives
$x_2(t) = 5t^4 + 4c_4t^3 + 3c_3t^2 + 2c_2t + c_1$

Solving for initial conditions,
we get,

$$c_0 = 0, c_1 = -1$$

Further control variable is calculated using the state equation which leads to

$$u(t) = 5t^4 + (4c_4 + 20)t^3 + (3c_3 + 12c_4)t^2 + (2c_2 \\ +6c_3)t + (c_1 + 2c_2)$$

which further leads to, $J$=

$$\frac{1}{2}\begin{pmatrix} c_2 & c_3 & c_4 \end{pmatrix}\begin{pmatrix} 34.52 & 37.69 & 39.79 \\ 37.69 & 44.82 & 50.15 \\ 39.79 & 50.15 & 58.44 \end{pmatrix}\begin{pmatrix} c_2 \\ c_3 \\ c_4 \end{pmatrix}$$

$$+\begin{pmatrix} c_2 & c_3 & c_4 \end{pmatrix}\begin{pmatrix} 13.31 \\ 25.62 \\ 39.50 \end{pmatrix}$$

Solving the above quadratic programming problem and minimizing $J$ with respect to $c_2, c_3, c_4$ gives

$$(c_2, c_3, c_4) = (-7.3436, 19.2045, -12.1543)$$

Putting these values in original equations we get the values as follows:

$$x_1^*(t) = t^5 - 12.1543t^4 + 19.2045t^3 - 7.3436t^2 - t;$$
$$x_2^*(t) = 5t^4 - 48.6172t^3 + 57.6135t^2 - 14.687t - 1;$$
$$u^*(t) = 5t^4 - 28.62t^3 - 88.238t^2 + 100.54t - 15.69;$$

Now we formulate the above problem as inverse optimal problem and compare results with our initial assumed values of $(\alpha_1, \alpha_2, \alpha_3)$.

$$\underset{\alpha_1,\alpha_2,\alpha_3}{\text{minimize}} \sum_{i=1}^{n}[(x_1^*(t_i) - x_1(t_i))^2 + (x_2^*(t_i) - x_2(t_i))^2 \\ + (u^*(t_i) - u(t_i))^2] \quad (2)$$

$$(c_2, c_3, c_4) \in \underset{c_2,c_3,c_4}{\text{argmin}}\ J$$

$$where,$$

$$\sum_{i=1}^{3}\alpha_i = 25.1 \quad (3)$$

and $J$ as defined in equation 1.
Carrying out the similar calculations, we get
$J = CA^{\intercal}B$, where

$$A = \begin{pmatrix} .1380 & 1.778 & 73.92 \\ -.250 & 1.333 & 37.33 \\ -.177 & 2.285 & 54.28 \\ -.130 & 3.000 & 115.0 \\ .3330 & 3.000 & 27.00 \\ .2500 & 4.000 & 64.00 \\ .2850 & 3.200 & 35.20 \\ .2000 & 1.333 & 9.330 \\ .1420 & 1.800 & 22.80 \\ .1110 & 2.285 & 47.08 \end{pmatrix} B = \begin{pmatrix} 1 \\ c_2 \\ c_3 \\ c_4 \\ c_2c_3 \\ c_3c_4 \\ c_2c_4 \\ c_2^2 \\ c_3^2 \\ c_4^2 \end{pmatrix}$$

$$C = \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 \end{pmatrix}$$

Please note that $n$ is the number of timestamps on which measurement is made in expression 2. Comparing our framework with standard bilevel optimization problem, notice that expression 2 is the upper level task with $(c_2, c_3, c_4) \in \underset{c_2,c_3,c_4}{\text{argmin}}\ J$ being the lower level task and equation 3 acting as lower level constraints. This formulation can also be compared with one which is defined in subsection 3.2 with,

$$\phi_1(\cdot) = x_1^2(t),\ \phi_2(\cdot) = x_2^2(t),\ \phi_3(\cdot) = u^2(t) \text{ and}$$

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u(t);$$

$\vartheta_M$ presents the measured data or data set as $x_1^*(t), x_2^*(t)$ and $u^*(t)$. MATLAB script was executed for $n = 100$ timestamps to get the fit on $t \in [0,1]$ with $t_i = 0.0, 0.01, \ldots, .51, .52, \ldots, 1.0$.
The numerical values calculated for upper level variables have been tabulated in table 1 with error function variation plotted in figure 3.

Table 1: Upper Level Variables after 700 Generations

| Execution | $\alpha_1$ | $\alpha_2$ | $\alpha_3$ |
|---|---|---|---|
| First | 15.0456 | 9.9551 | 0.0993 |
| Second | 14.9873 | 10.0125 | 0.1002 |
| Third | 14.9995 | 10.0005 | 0.1001 |
| Fourth | 15.0162 | 9.9841 | 0.0997 |
| Fifth | 15.0207 | 9.9796 | 0.0997 |
| Average | 15.0386 | 9.9864 | .0998 |

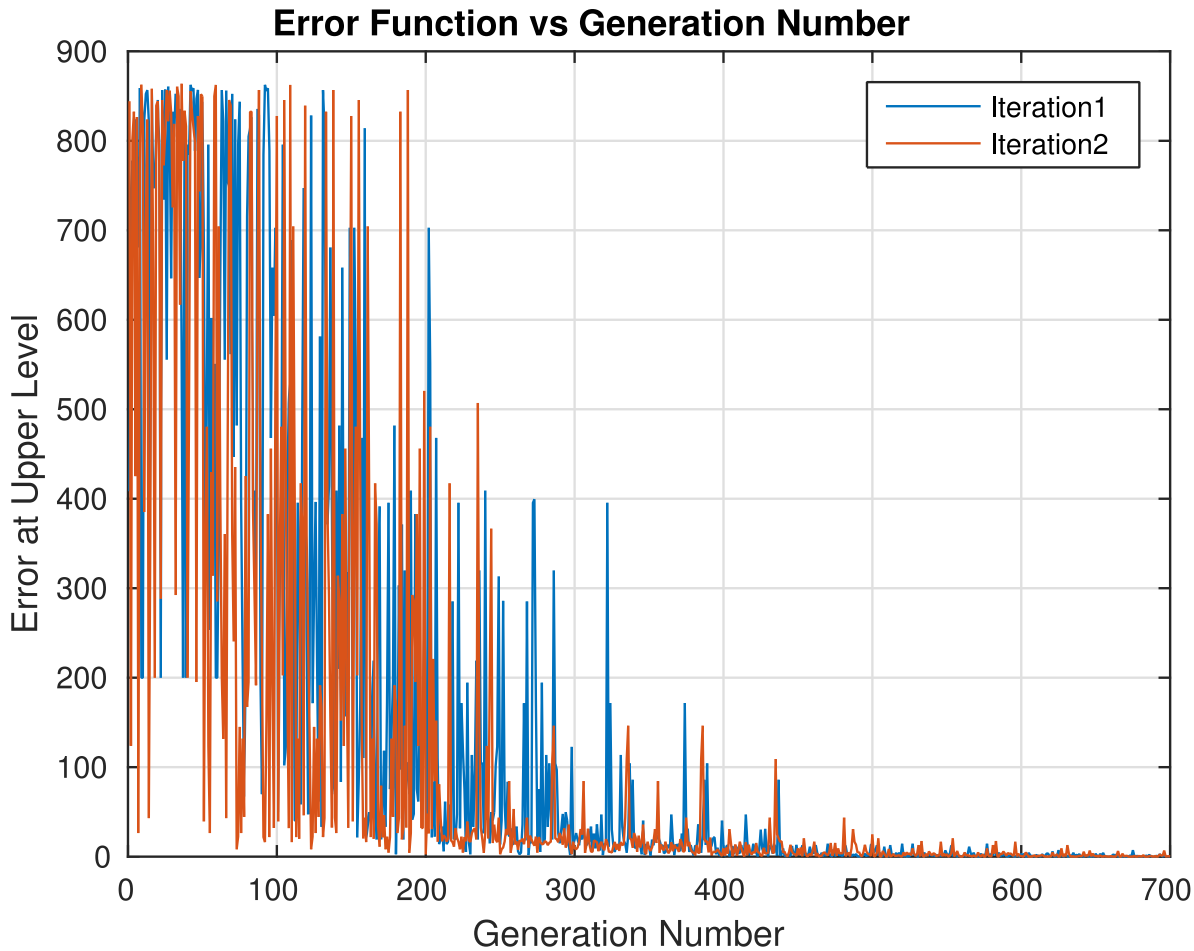


Figure 3: Error function at upper level variation against generation number. Clearly, Squared error vanishes and converges to 0 after few generations advocating a strong fit between calculated values and actual values.

#### 4.1.2 Example 2

This example problem depicts a non-linear system adopted from optimal control problem described in [45].

$$\text{Find } u^*(t) \text{ that minimizes}$$

$$J = \int_0^1 [\alpha_1 x_1(t) + \alpha_2 x_2(t) + \alpha_3 u(t)]dt \tag{4}$$

subject to

$$\dot{x}_2 = -2x_1^2 - x_2 + 2u,$$
$$\dot{x}_1 = -2x_1 + x_2,$$

and

$$(\alpha_1, \alpha_2, \alpha_3) = (1, 2, 1)$$

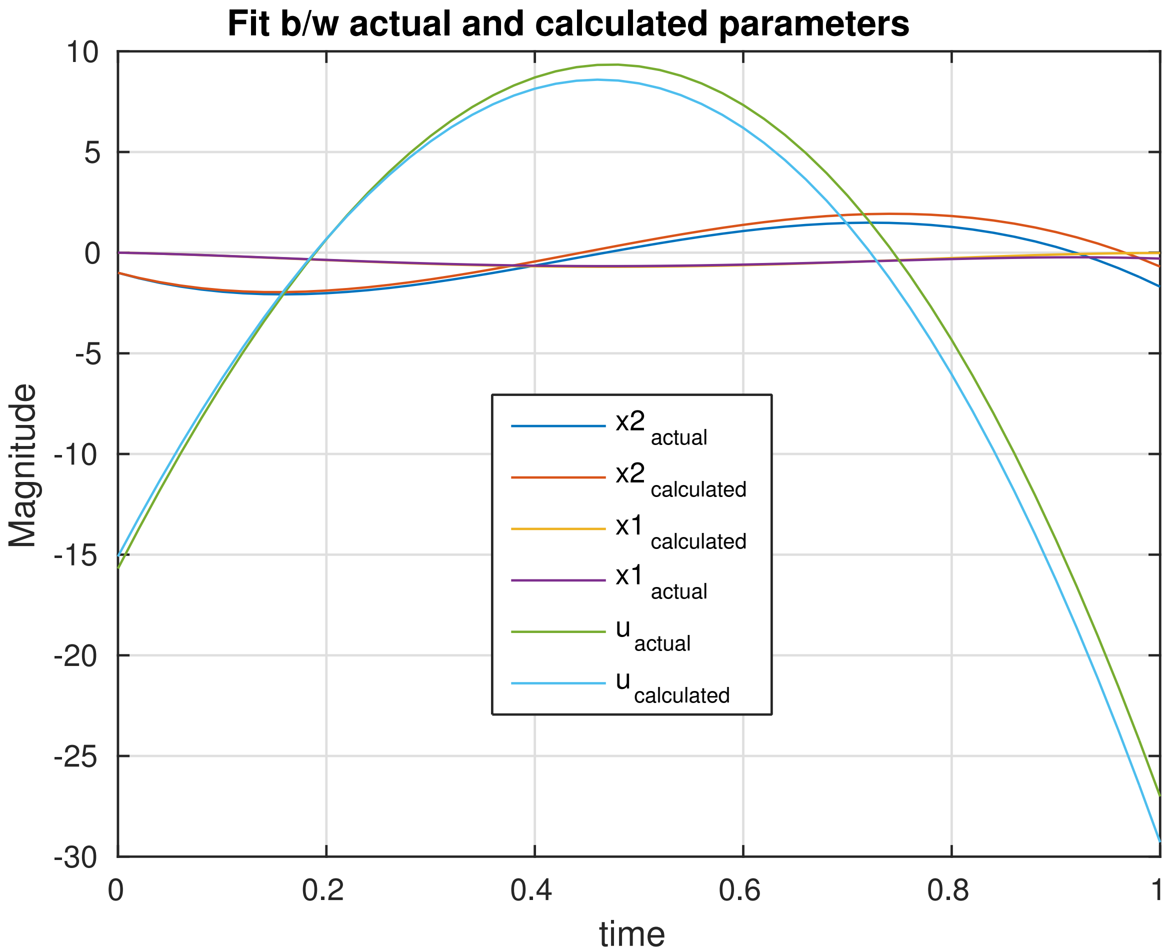


Figure 4: Fitness between approximation polynomials based upon actual and calculated parameters $(c_2, c_3, c_4)$ after 700 generations for example problem 4.1.1

Since the limits are from 0 to 1 so let us we approximate $x_1(t)$ by a harmonic function which has 4 degrees of freedom,

$$x_1(t) = \mathrm{A}_1\ \sin(\omega_1\, t) + \mathrm{A}_2\ \sin(\omega_2\, t)$$

Then, $\dot{x}_2(t)$ is given by the expression,

$$\begin{aligned} x_2(t) = 2\,\mathrm{A}_1\ \sin(\omega_1\, t) + 2\,\mathrm{A}_2\ \sin(\omega_2\, t) + \\ \mathrm{A}_1\,\omega_1\ \cos(\omega_1\, t) + \mathrm{A}_2\,\omega_2\ \cos(\omega_2\, t) \end{aligned}$$

Further control variable is calculated using the state equation which leads to $u(t) =$

$$\begin{aligned} \sin(\omega_1\, t)\ \left(A_1 - \frac{A_1\,{\omega_1}^2}{2}\right) + \sin(\omega_2\, t)\ \left(A_2 - \frac{A_2\,{\omega_2}^2}{2}\right) \\ +(A_1\ \sin(\omega_1\, t) + A_2\ \sin(\omega_2\, t))^2 + \frac{3\,A_1\,\omega_1\ \cos(\omega_1\, t)}{2} \\ + \frac{3\,A_2\,\omega_2\ \cos(\omega_2\, t)}{2} \end{aligned} \tag{5}$$

which further leads to,

$$
\begin{aligned}
J = {} & \frac{6\,A_1}{\omega_1} - \frac{A_2\,\omega_2}{2} - \frac{A_1\,\omega_1}{2} + \frac{6\,A_2}{\omega_2} + \frac{{A_1}^2}{2} + \frac{{A_2}^2}{2} + \\
& \frac{7\,A_1\,\sin(\omega_1)}{2} + \frac{7\,A_2\,\sin(\omega_2)}{2} - \frac{6\,A_1\,\cos(\omega_1)}{\omega_1} - \\
& \frac{6\,A_2\,\cos(\omega_2)}{\omega_2} + \frac{A_1\,\omega_1\,\cos(\omega_1)}{2} + \frac{A_2\,\omega_2\,\cos(\omega_2)}{2} - \\
& \frac{{A_1}^2\,\cos(\omega_1)\,\sin(\omega_1)}{2\,\omega_1} - \frac{{A_2}^2\,\cos(\omega_2)\,\sin(\omega_2)}{2\,\omega_2} - \\
& \frac{2\,A_1\,A_2\,\omega_1\,\cos(\omega_1)\,\sin(\omega_2)}{{\omega_1}^2 - {\omega_2}^2} + \\
& \frac{2\,A_1\,A_2\,\omega_2\,\cos(\omega_2)\,\sin(\omega_1)}{{\omega_1}^2 - {\omega_2}^2}
\end{aligned}
$$

It is clearly visible that $J$ is symmetric with respect to pair $(\omega_1, \omega_2)$ and pair $(A_1, A_2)$. $J$ was minimized using *fmincon*[?] with to get $(\omega_1, \omega_2, A_1, A_2) = (4.5554, 4.5553, 1.4347, 3.2935)$ with following restrictions imposed $3 \leq \omega_1 \leq 5, 3 \leq \omega_2 \leq 5, 1 \leq A_1 \leq 4, 3 \leq A_2 \leq 4$. Putting these values in original equations we get the values as follows:

$$
\begin{aligned}
x_1^*(t) &= 1.4347\,\sin(4.5554\,t) + 3.2935\,\sin(4.5553\,t) \\
x_2^*(t) &= 15.003\,\cos(4.5553\,t) + 6.536\,\cos(4.5554\,t) + \\
& \quad 6.587\,\sin(4.5553\,t) + 2.869\,\sin(4.5554\,t)
\end{aligned}
$$

$u^*(t)$ can be calculated by putting results in equation 5.
Again we formulated above problem as inverse optimal problem as described already in 4.1.1 and formulation can easily be compared with one defined in subsection 3.2.
The numerical values calculated for upper level variables have been tabulated in table 2 with error function variation plotted in figure 5.

Table 2: Upper Level Variables after 800 Generations

| Execution | $\alpha_1$ | $\alpha_2$ | $\alpha_3$ |
|---|---|---|---|
| First | 1.0021 | 1.9985 | 0.9994 |
| Second | 0.9995 | 2.0012 | 0.9993 |
| Third | 1.0012 | 1.9985 | 1.0003 |
| Fourth | 0.9998 | 1.9978 | 1.0024 |
| Fifth | 0.9986 | 2.0023 | 0.9991 |

#### 4.1.3 Example 3

Now we solve a real world application of optimal control theory where it can be used to design robot trajectories which can imitate human beings. Many researchers dealt this problem in past

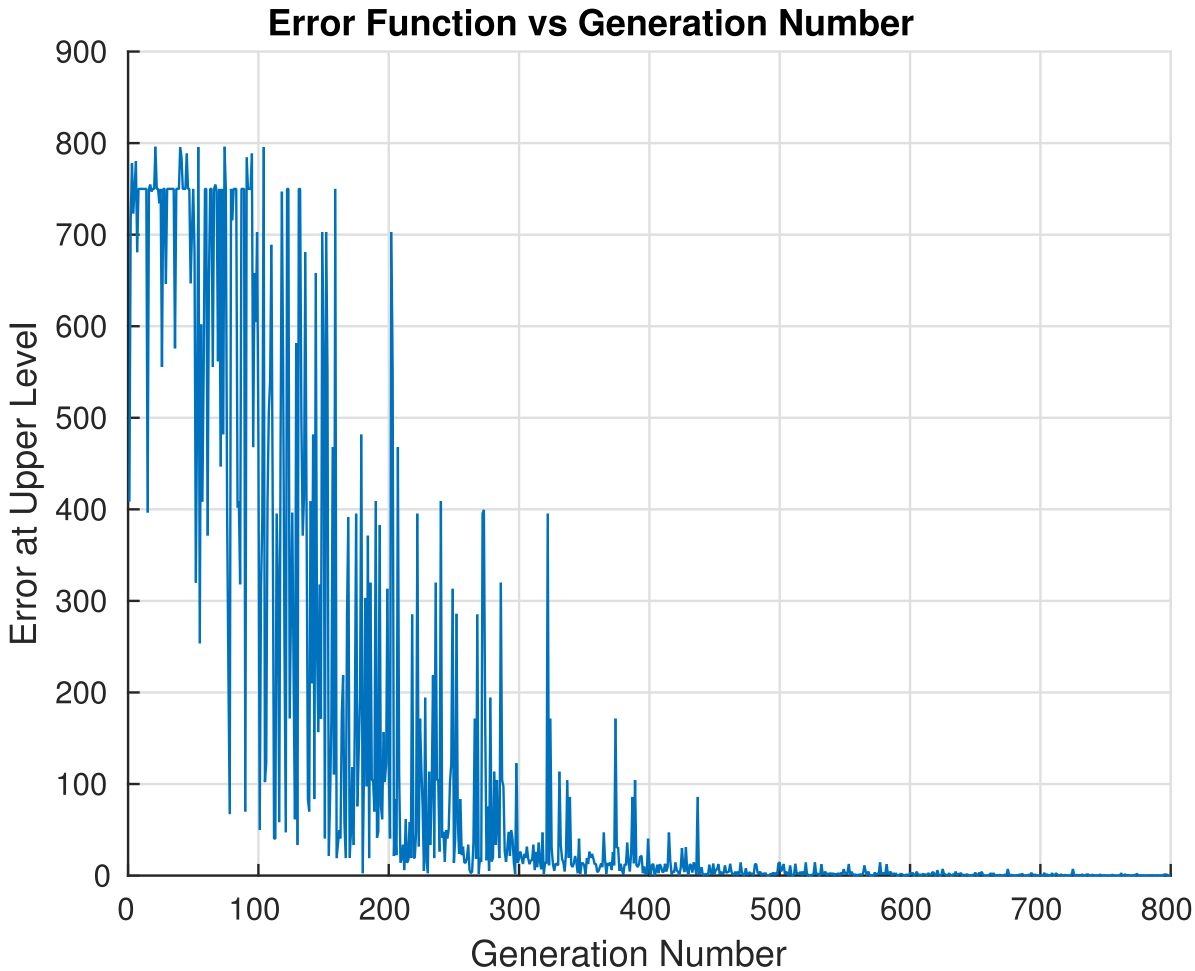


Figure 5: Error function at upper level variation against generation number.

[50, 53, 54]. Our problem has been inspired from what has been discussed in [54]. A common assumption of many approaches analyzing human motion is that humans try to minimize an unknown cost function while doing everyday manipulation tasks. This assumption also forms the basis for the strategy discussed here. It has to be noticed that the main purpose of this approach is to describe human behavior based on physical models rather than to explain it from a biological or psychological perspective [55]. In daily life a huge potential exists to assist and disencumber humans in fulfilling their tasks, by providing service or care robots. A difficult task in building adaptive and autonomous robots is generating task-specific robot motions which fit naturally in a human everyday environment. Our contribution is a framework which is able to represent and generate motions based on an optimal combination of physically inspired principles obtained by analyzing human motions recorded in an unconstrained observation setup. Having all of these in mind, our proposed method works on the principal of BLEAQ. While cost function formulation has been adopted from [50] which leads to a bilevel task as follows:

We took the following basic formulation of the objective function as a combined weighted minimization of total time, the integrated squares of the three acceleration components, and the integrated squared difference of body orientation angle and direction towards the goal $\Psi(t)$ as described in

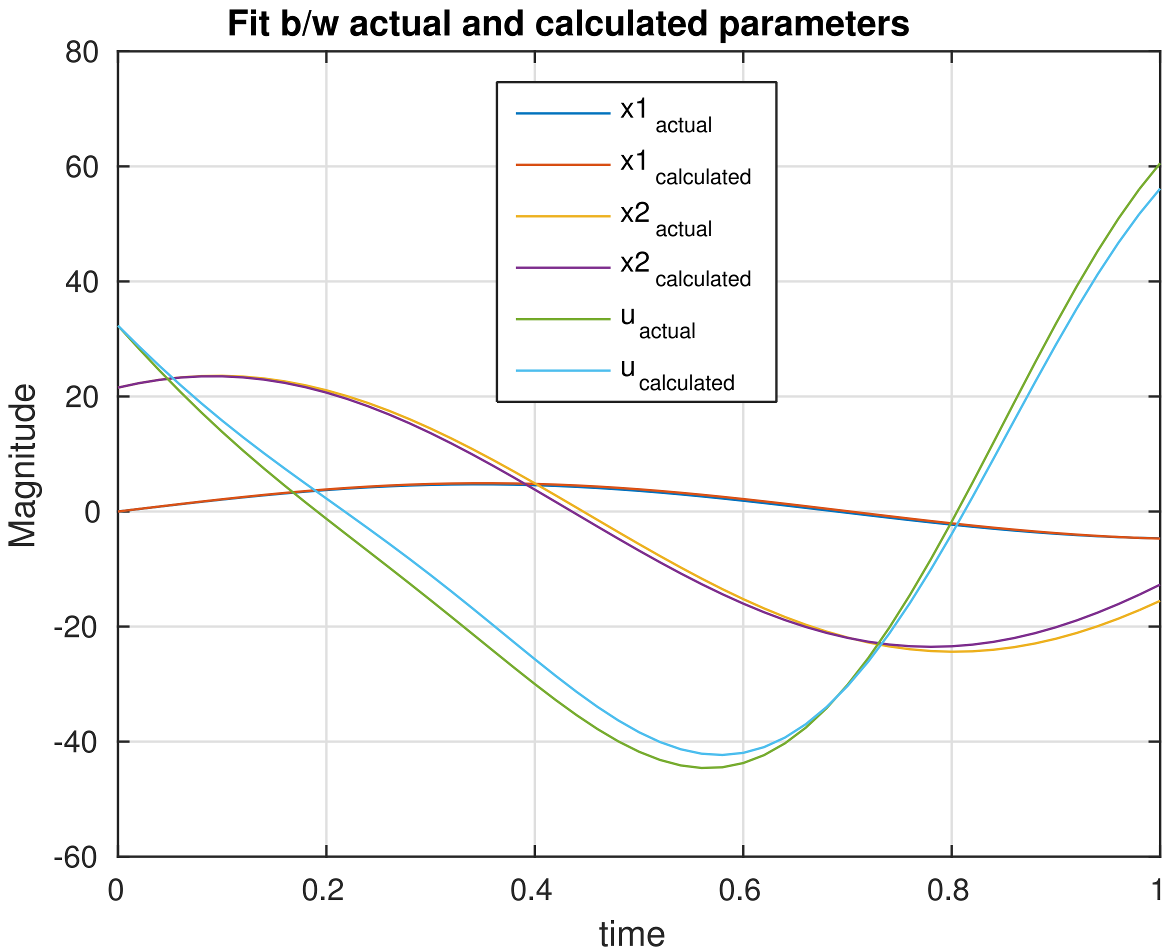


Figure 6: Fitness between approximated harmonic functions based upon actual and calculated parameters $(\omega_1, \omega_2, A_1, A_2)$ after 800 generations for example problem 4.1.2

[50]:

$$= \int_0^T \Big[ \sum_0^4 \alpha_i \phi_i \Big( z(t), u(t) \Big) \Big] dt$$

$$= \int_0^T [\alpha_0 + \alpha_1 u_1(t)^2 + \alpha_2 u_2(t)^2 +$$

$$\alpha_3 u_3(t)^2 + \alpha_4 \Psi(z(t), z_e)^2] dt \tag{6}$$

$$= \alpha_0 T + \alpha_1 \int_0^T u_1(t)^2 dt + \alpha_2 \int_0^T u_2(t)^2 dt +$$

$$\alpha_3 \int_0^T u_3(t)^2 dt + \alpha_4 \int_0^T \Psi(z(t), z_e)^2 dt \tag{7}$$

subject to

$$\Psi(z(t), z_e) = \frac{y_e - y(t)}{x_e - x(t)} - \theta(t) \text{ and}$$

$$-\pi \leq \Psi(z(t), z_e) \leq \pi$$


Figure 7: Human performing reaching motions

A detailed description of the variables and parameters formulated above can be found in [50, 54]. We implemented our algorithm on TUM kitchen dataset. Detailed description of the dataset can be found in [54, 56].

Value of cofficients has been calculated as $(\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4)$=(1.86, 2.53, 3.33, 1.36, 5.3) Right arm path (sequence of coordinates) has been plotted in figure 8. Trajectories obtained by using our cofficients closely follow the actual data set values showing the robustness and applicability of our method.

# 5 Conclusions and Future Work

In this paper, we have solved inverse optimal control problems using a bilevel optimization technique. A recently proposed technique (BLEAQ) has been applied on a couple of toy examples and a real world problem. The results suggest that the technique is able to successfully handle all the problems, and can therefore be useful in solving more complex inverse control problems. The problems need not adhere to continuity and differentiability assumptions for the algorithm to be applicable. One of the important applications of inverse optimal control is in human imitation where robots are being sought which can perfectly imitate humans in performing tasks. While performing an act humans minimize an unknown cost function, and researchers spanning from psychologists to mathematicians are engaged in figuring out the optimal criteria. Bilevel optimization can be a useful approach for determining the optimal cost function, which is in an important direction on which we would like to extend our research. Development of efficient algorithms for bilevel optimization and accurate human motion tracking techniques would enable more accurate models of human motion and performance and thus transferring these models to humanoids which can lead to greater efficiency of robots.

# References

[1] L. S. Pontryagin, *Mathematical theory of optimal processes*, 1962.

[2] R. E. Bellman, *Dynamic Programming.* Princeton, NJ, 1957.

[3] W. Zhang and H.-p. Ma, "Chebyshev-legendre method for discretizing optimal control problems," *Journal of Shanghai University (English Edition)*, vol. 13, pp. 113–118, 2009.

[4] B. Kafash, "Numerical solution of nonlinear optimal control problems based on state parametrization," *Iranian Journal of Science and Technology (Sciences)*, vol. 36, pp. 331–340, 2014.

[5] H. v. Stackelberg, *The theory of the market economy.* New York: Oxford University Press, 1952.

[6] L. Brotcorne, M. Labbe, P. Marcotte, and G. Savard, "A bilevel model for toll optimization on a multicommodity transportation network," *Transportation Science*, vol. 35, no. 4, pp. 345–358, 2001.

[7] A. Migdalas, "Bilevel programming in traffic planning: Models, methods and challenge," *Journal of Global Optimization*, vol. 7, no. 4, pp. 381–405, 1995.

[8] Y. Yin, "Multiobjective bilevel optimization for transportation planning and management problems," *Journal of advanced transportation*, vol. 36, no. 1, pp. 93–105, 2002.

[9] A. Sinha, P. Malo, and K. Deb, "Transportation policy formulation as a multi-objective bilevel optimization problem," in *2015 IEEE Congress on Evolutionary Computation (CEC-2015).* IEEE Press, 2015.

[10] M. A. Amouzegar and K. Moshirvaziri, "Determining optimal pollution control policies: An application of bilevel programming," *European Journal of Operational Research*, vol. 119, no. 1, pp. 100–120, 1999.

[11] A. Sinha, P. Malo, A. Frantsev, and K. Deb, "Multi-objective stackelberg game between a regulating authority and a mining company: A case study in environmental economics," in *2013 IEEE Congress on Evolutionary Computation (CEC-2013).* IEEE Press, 2013.

[12] M. Bostian, A. Sinha, G. Whittaker, and B. Barnhart, "Incorporating data envelopment analysis solution methods into bilevel multi-objective optimization," in *2015 IEEE Congress on Evolutionary Computation (CEC-2015).* IEEE Press, 2015.

[13] W. Smith and R. Missen, *Chemical Reaction Equilibrium Analysis: Theory and Algorithms.* John Wiley & Sons, New York, 1982.

[14] P. A. Clark and A. W. Westerberg, "Bilevel programming for steady-state chemical process designi. fundamentals and algorithms," *Computers & Chemical Engineering*, vol. 14, no. 1, pp. 87–97, 1990.

[15] A. U. Raghunathan and L. T. Biegler, "Mathematical programs with equilibrium constraints (mpecs) in process engineering," *Computers & Chemical Engineering*, vol. 27, no. 10, pp. 1381–1392, 2003.

[16] J. Herskovits, A. Leontiev, G. Dias, and G. Santos, "Contact shape optimization: a bilevel programming approach," *Structural and multidisciplinary optimization*, vol. 20, no. 3, pp. 214–221, 2000.

[17] S. Christiansen, M. Patriksson, and L. Wynter, "Stochastic bilevel programming in structural optimization," *Structural and multidisciplinary optimization*, vol. 21, no. 5, pp. 361–371, 2001.

[18] M. P. Bendsoe, "Optimization of structural topology, shape, and material," Tech. Rep., 1995.

[19] J. Bracken and J. T. McGill, "Defense applications of mathematical programs with optimization problems in the constraints," *Operations Research*, vol. 22, no. 5, pp. 1086–1096, 1974.

[20] E. Israeli and R. K. Wood, "Shortest-path network interdiction," *Networks*, vol. 40, no. 2, pp. 97–111, 2002.

[21] G. Brown, M. Carlyle, J. Salmerón, and K. Wood, "Defending critical infrastructure," *Interfaces*, vol. 36, no. 6, pp. 530–544, 2006.

[22] M. P. Scaparra and R. L. Church, "A bilevel mixed-integer program for critical infrastructure protection planning," *Computers & Operations Research*, vol. 35, no. 6, pp. 1905–1923, 2008.

[23] W. Bialas and M. Karwan, "Two-level linear programming," *Management Science*, vol. 30, pp. 1004–1020, 1984.

[24] Y. Chen and M. Florian, "On the geometry structure of linear bilevel programs: a dual approach," Centre de Recherche sur les Transports, Tech. Rep. CRT-867, 1992.

[25] H. Tuy, A. Migdalas, and P. Värbrand, "A global optimization approach for the linear two-level program," *Journal of Global Optimization*, vol. 3, pp. 1–23, 1993.

[26] J. Bard and J. Falk, "An explicit solution to the multi-level programming problem," *Computers and Operations Research*, vol. 9, pp. 77–100, 1982.

[27] J. Fortuny-Amat and B. McCarl, "A representation and economic interpretation of a two-level programming problem," *Journal of the Operational Research Society*, vol. 32, pp. 783–792, 1981.

[28] J. Bard and J. Moore, "A branch and bound algorithm for the bilevel programming problem," *SIAM Journal on Scientific and Statistical Computing*, vol. 11, pp. 281–292, 1990.

[29] T. Edmunds and J. Bard, "Algorithms for nonlinear bilevel mathematical programming," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, pp. 83–89, 1991.

[30] F. Al-Khayyal, R. Horst, and P. Pardalos, "Global optimization of concave functions subject to quadratic constraints: an application in nonlinear bilevel programming," *Annals of Operations Research*, vol. 34, pp. 125–147, 1992.

[31] G. Savard and J. Gauvin, "The steepest descent direction for the nonlinear bilevel programming problem," *Operations Research Letters*, vol. 15, pp. 275–282, 1994.

[32] L. Vicente, G. Savard, and J. Júdice, "Descent approaches for quadratic bilevel programming," *Journal of Optimization Theory and Applications*, vol. 81, pp. 379–399, 1994.

[33] E. Aiyoshi and K. Shimizu, "Hierarchical decentralized systems and its new solution by a barrier method," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, pp. 444–449, 1981.

[34] ——, "A solution method for the static constrained Stackelberg problem via penalty method," *IEEE Transactions on Automatic Control*, vol. 29, pp. 1111–1114, 1984.

[35] G. Liu, J. Han, and S. Wang, "A trust region algorithm for bilevel programing problems," *Chinese science bulletin*, vol. 43, no. 10, pp. 820–824, 1998.

[36] P. Marcotte, G. Savard, and D. Zhu, "A trust region algorithm for nonlinear bilevel programming," *Operations research letters*, vol. 29, no. 4, pp. 171–179, 2001.

[37] B. Colson, P. Marcotte, and G. Savard, "A trust-region method for nonlinear bilevel programming: algorithm and computational experience," *Computational Optimization and Applications*, vol. 30, no. 3, pp. 211–227, 2005.

[38] R. Mathieu, L. Pittard, and G. Anandalingam, "Genetic algorithm based approach to bi-level linear programming," *Revue française d'automatique, d'informatique et de recherche opérationnelle. Recherche opérationnelle*, vol. 28, no. 1, pp. 1–21, 1994.

[39] Y. Yin, "Genetic algorithm based approach for bilevel programming models," *Journal of Transportation Engineering*, vol. 126, no. 2, pp. 115–120, 2000.

[40] Y. Wang, Y. C. Jiao, and H. Li, "An evolutionary algorithm for solving nonlinear bilevel programming based on a new constraint-handling scheme," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 32, no. 2, pp. 221–232, 2005.

[41] Y. Wang, H. Li, and C. Dang, "A new evolutionary algorithm for a class of nonlinear bilevel programming problems and its global convergence," *INFORMS Journal on Computing*, vol. 23, no. 4, pp. 618–629, 2011.

[42] A. Sinha, P. Malo, and K. Deb, "Efficient evolutionary algorithm for single-objective bilevel optimization," *arXiv preprint arXiv:1303.3901*, 2013.

[43] ——, "An improved bilevel evolutionary algorithm based on quadratic approximations," in *2014 IEEE Congress on Evolutionary Computation (CEC-2014)*. IEEE Press, 2014, pp. 1870–1877.

[44] J. S. Angelo, E. Krempser, and H. J. Barbosa, "Differential evolution assisted by a surrogate model for bilevel programming problems," in *Evolutionary Computation (CEC), 2014 IEEE Congress on*. IEEE, 2014, pp. 1784–1791.

[45] F. Lin, *Robust control design: an optimal control approach*. John Wiley & Sons, 2007, vol. 18.

[46] R. Kalman, "When is a linear control system optimal?" *Trans AMSE J Basic Eng*, pp. 51–60, 1964.

[47] T. E. Dvijotham K, "Inverse optimal control with linearly-solvable mdps," 2010.

[48] H. Deng and M. Krstić, "Stochastic nonlinear stabilizationii: inverse optimality," *Systems & Control Letters*, vol. 32, no. 3, pp. 151–159, 1997.

[49] M. Ziebart, “Maximum entropy inverse reinforcement learning,” *AAAI*, pp. 1433–1438, 2008.

[50] K. Mombaur, A. Truong, and J.-P. Laumond, “From human to humanoid locomotion1̣an inverse optimal control approach,” *Autonomous robots*, vol. 28, no. 3, pp. 369–383, 2010.

[51] B. Kafash, A. Delavarkhalafi, and S. Karbassi, “A numerical approach for solving optimal control problems using the boubaker polynomials expansion scheme,” *Journal of Interpolation and Approximation in Scientific Computing*, vol. 2014, pp. 1–18, 2014.

[52] H. H. Mehne and A. H. Borzabadi, “A numerical method for solving optimal control problems using state parametrization,” *Numerical Algorithms*, vol. 42, no. 2, pp. 165–169, 2006.

[53] S. Albrecht, M. Sobotka, and M. Ulbrich, “A bilevel optimization approach to obtain optimal cost functions for human arm-movements,” *Fakultät für Mathematik, TU München*, 2010.

[54] S. Albrecht, K. Ramirez-Amaro, F. Ruiz-Ugalde, D. Weikersdorfer, M. Leibold, M. Ulbrich, and M. Beetz, “Imitating human reaching motions using physically inspired optimization principles,” in *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*. IEEE, 2011, pp. 602–607.

[55] Engelbrecht, “Minimum principles in motor control,” *J. Mathematical Psychology*, vol. 45, no. 3, pp. 497–542, 2001.

[56] M. Tenorth, J. Bandouch, and M. Beetz, “The TUM Kitchen Data Set of Everyday Manipulation Activities for Motion Tracking and Action Recognition,” in *IEEE International Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS), in conjunction with ICCV2009*, 2009.

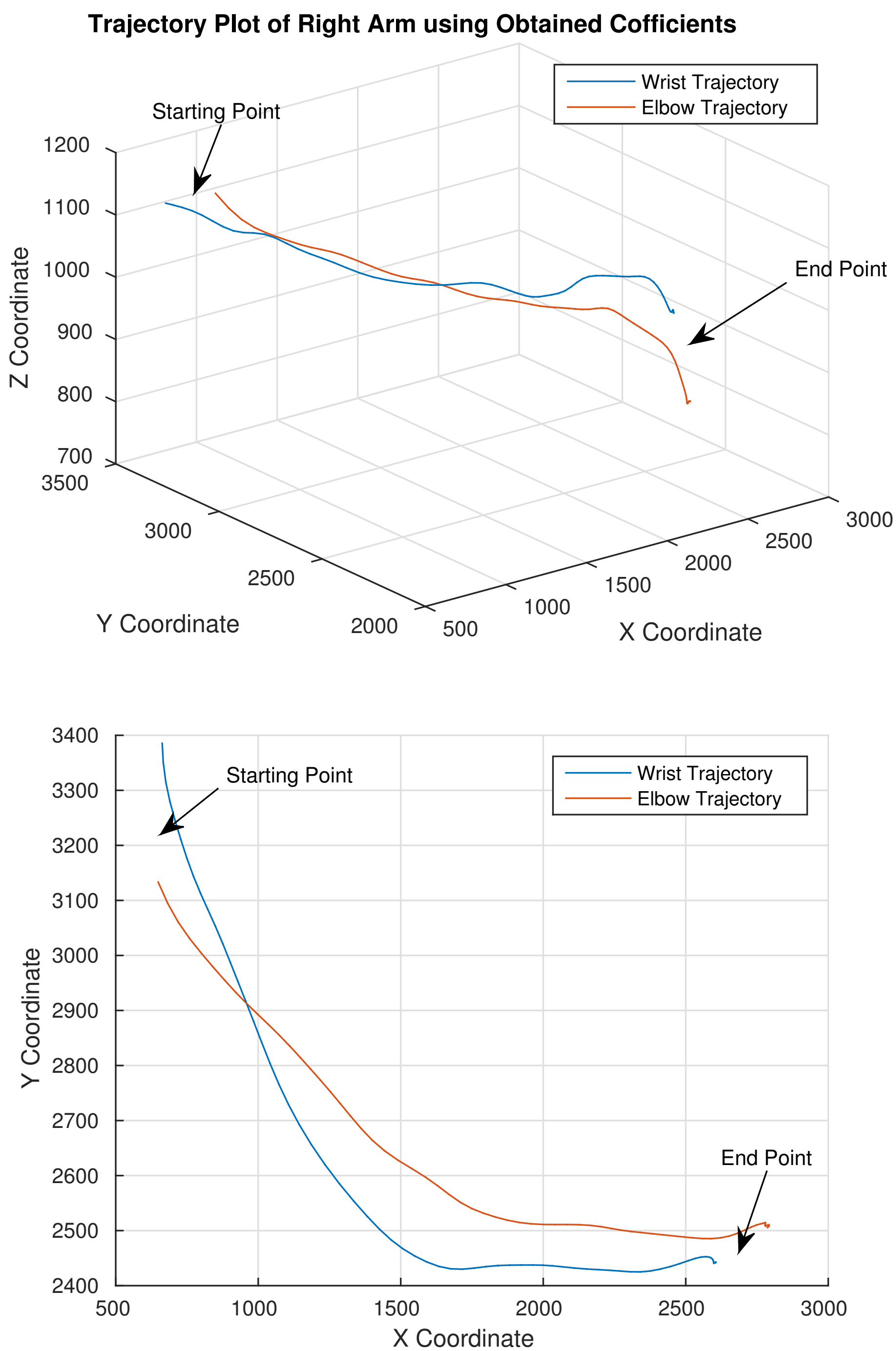


Figure 8: Right arm trajectory plot in reaching motions for 4.1.3