Development of an Adaptive-Optimal Multi-Objective Optimization Algorithm

Ali Abdollahi* and Girish Chowdhary†

In this paper, we consider the problem of synthesizing online optimal flight controllers, in the presence of multiple objectives. The problem is cast as an adaptive Multi-Objective Optimization (MO-Op) flight control problem, in which a control policy is sought that attempt to optimize over multiple, sometimes conflicting objectives. A solution strategy utilizing Gaussian Process (GP) based adaptive-optimal control is presented, in which the system uncertainties are learned with an online updated budgeted GP. The mean of the GP is used to feedback linearize the system and reference model shaping is utilized for optimization. To make the MO-Op problem online realizable, a relaxation strategy that poses some objectives as adaptively updated soft constraints is proposed. The strategy is validated on a nonlinear roll dynamics model with simulated state-dependent flexible-rigid mode interaction.

I. Introduction

Fuel efficiency and environmental compatibility can both be significantly improved by utilizing light weight and high-aspect ratio wing design.[1] However, using lightweight or high aspect ratio constructions leads to structures with flexible modes, which can lead to increased component wear or decreased ride comfort. Hence, in order to leverage lightweight or high-aspect ratio construction, the aircraft control system must be designed to optimize over multiple, sometimes conflicting objectives, such as maintaining a wing profile that minimizes drag, alleviating gust loading, dampening out flexible modes to improve ride comfort, and ensuring that the aircraft has good handling qualities when tracking pilot commands.

Decision making problems that optimize over multiple, sometimes conflicting, objectives simultaneously are generally known as Multi-Objective Optimization (MO-Op) problems. A characteristic of these types of problems is that there does not always exist a unique solution, rather a number of mathematically equally good solutions called Pareto optimal solutions.[2,3] The set of feasible solutions is not explicitly known in advance for these problems but it can be restricted by constraint functions. There are two key challenges in developing Multi-Objective Optimization (MO-Op) flight control techniques. Firstly, the dynamic model of the aircraft may not always be known in advance, especially in the presence of intricate coupling between flexible and rigid body modes. Furthermore, even if such a model is known or can be learned using online data, a feasible optimal solution might be difficult to be found online in the presence of computational constraints.

The main goal of this paper is to describe ongoing work towards a framework that can be utilized for adaptive MO-Op flight control by marrying together learning-focused model reference adaptive control with online Model Predictive Control based optimization framework. The critical elements of this architecture are a learning engine, an online realizable MO-Op strategy, and a feedback structure that ensures the system states are stable during adaptation. We utilize a Gaussian Process (GP) based MPC architecture for online learning of the modeling uncertainty. Furthermore, we show that when optimizing to achieve good command tracking while minimizing oscillations due to exciting flexible modes, the MO-Op problem can be relaxed by posing that MO-Op objective as adaptively updated soft state constraint. This makes the problem practically realizable online. The entire architecture is realized through the reference model shaping MPC architecture introduced in [4, 5].

A. Related Work

Model Predictive Control (MPC), also referred to as Receding Horizon Control and Moving Horizon Optimal Control, has been widely adopted in industry as an effective means to deal with multivariable constrained control problems.[6,7]

*Graduate Research Assistant, Mechanical and Aerospace Engineering, Oklahoma State University, and Student Member.
†Assistant Professor, Mechanical and Aerospace Engineering, Oklahoma State University, and AIAA Member.

With advent in compact computing resources, authors have recently proposed and employed MPC architectures for real-time control of aerospace vehicles.[8] However, the performance of MPC can depend on how the accuracy of the employed predictive model of the system dynamics is. Several authors have proposed learning based MPC algorithms,[9–11] however the presence of learning transients typically prevents a non-conservative solution to be formed. The CL-MPC architecture addressed this issue by bringing together learning based MRAC and MPC and utilizes an online threshold test to switch between the two.[12,13] However, CL-MPC is developed in the deterministic setting and it requires online optimization of the feedback linearzed system model. The latter represents one of the main challenges in implementing MPC based architectures: guaranteeing computational feasibility of obtaining an optimal solution online. Muhlegg et al. in [4], this problem is tackled by solving offline the optimal control problem for a linear reference model that the adaptive controller is guaranteed to track. Inspired by [4], our main contributions include the use of Bayesian Non-Parametric (BNP) Gaussian Process models of the uncertainty that require very few assumptions about the unknown system dynamics a-priori.

Multi-objective optimization has been studied for several decades based on the theoretical background established in 1890's.[14] In a restricted sense, the Linear Quadratic Regulator, in which the control objective is to both reduce the tracking error and minimize required control, can be considered as a MO-Op problem posed by linearly combining multiple control objectives. More general formulations and solution approaches to MO-Op have been founded on the theory of mathematical programming. The key challenge with MO-Op is that there may exist mathematically equally good solutions, known as Pareto optimal solutions. Many MO-Op methods rely on the human in the loop to discriminate between these solutions.[3] In general, there has been very little work in utilizing MO-Op in an online setting, and we are not aware of any work on optimization based MO-Op in flight control settings. Our main contribution in this area is to show that a class of MO-Op flight control problems can be relaxed by posing some of the control objectives as online updated soft constraints.

II. Problem Formulation

We begin by describing the class of switching nonlinear dynamical systems our approach is applicable to. Let $x = [x_1 \ x_2]^\top \in D_x \subset \mathbb{R}^n$ be defined as the state vector of the nonlinear system where D_x is a subset of $\mathbb{R}^n$. Let $u(t) \in \mathbb{R}^m$ be an admissible control input to the system, then the class of dynamical systems considered here have the form:

$$\dot{x}(t) = Ax(t) + Bu(t) + B\Delta_i(x(t)). \tag{1}$$

where the state and input matrices $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are the linear known components, and $\Delta_i(x(t)) \in \mathbb{R}^m$ is the unknown, stochastic, and switching component in the system with the switching index i. The solution to (1) induces a nonstationary continuous time Markovian process. The process is nonstationary, i.e. it does not have a time invariant generating distribution, because the stochastic component Δ_i switches. We assume that the system (A,B) is fully controllable and the control input $u(t) \in \mathbb{R}^m$ is constrained due to limited control energy available to the system. It is further assumed that Δ_i is a stochastic process, and in particular a GP. Here we leverage the idea of modeling stochastic uncertainties $\Delta(x)$ in MRAC using Gaussian Processes.[15] That is,

$$\Delta_i(x) \sim \mathcal{GP}(m_i(x), K_i(x,x')), \tag{2}$$

where $m(\cdot)$ is the mean function of the GP and $K(\cdot,\cdot)$ is a real valued covariance function. Note that the mean function of a GP is globally Lipschitz continuous.

The problem we are interested in solving is tracking the states of a reference model x_{rm} in presence of the nonlinear uncertainty Δ_i and minimizing the oscillation in our reference model as the second objective in MO-Op problem. Let $x_{rm}(t) \in D_x \subset \mathbb{R}^n$ be the state of the reference model which characterizes the desired response of the system. The dynamics of the reference model are assumed to take on the form:

$$\dot{x}_{rm}(t) = A_{rm}x_{rm}(t) + B_{rm}u_{rm}(x(t), r(t)). \tag{3}$$

where $A_{rm} \in \mathbb{R}^{n\times n}$ and $B_{rm} \in \mathbb{R}^{n\times m}$ and the reference model input $u_{rm} \in \mathbb{R}^m$. The objective is to design a controller which can achieve a closed-loop performance such that the plant model defined by (1) tracks the reference model given by (3) while minimizing the oscillation. Therefore a control law with a feedback term defined as $u_{fb} = K_m^\top x; K \in \mathbb{R}^{n\times m}$, a linear feedforward term, $u_{ff} = K_b^\top r; K_b \in \mathbb{R}^{m\times m}$, and an adaptive element, u_{ad} is chosen to have the desired behavior.

$$u(t) = u_{ff}(t) + u_{fb}(t) - u_{ad}(t) \tag{4}$$

The gain matrices K_m and K_r are chosen such that $A_{rm} = A + BK_m^\top$ and $B_{rm} = BK_b$ [See (27)]. Define the tracking error as $e(t) = x(t) - x_{rm}(t)$. Then by using (1) and (3) and then substituting (4) into (1), we can obtain the tracking error dynamics as:

$$\dot{e}(t) = A_{rm}e(t) + B_{rm}(\Delta(x(t)) - u_{ad}(t)). \tag{5}$$

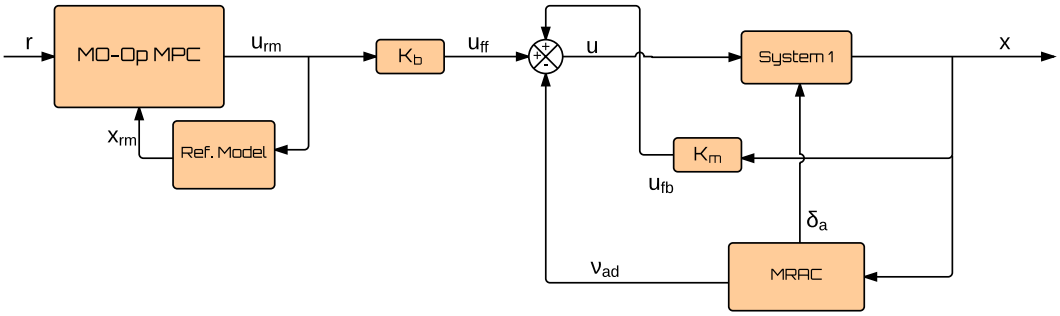


Figure 1. The proposed solution architecture combines two key elements: a reference model shaping MPC and a GP-MRAC based adaptive reference tracking architecture.

We are now in a position to present an overview of the presented architecture shown in Figure 1. The architecture consists of two key elements, a reference command shaping MO-Op MPC and GP-MRAC based adaptive control. The synthesis of u_{ad} is accomplished using the GP-MRAC method.[16] However, GP-MRAC is a tracking architecture that tracks a given reference command, it does not guarantee the optimality of the reference command, especially in presence of state and input constraints. To accommodate this, we utilize the MPC based reference command shaping technique proposed by Muhlegg et al.[4] and modify it such that we could incorporate multiple objectives in there. Hence, in our architecture, the reference command is shaped using MO-Op MPC on the reference model (3) in presence of state and input constraints optimizing over two objectives i.e. tracking error and oscillation. The optimized reference command is tracked by the GP-MRAC which simultaneously tracks the reference commands and learns a model of the underlying uncertainty Δ_i.

III. Methods

A. Multi-Objective Optimization (MO-Op)

The multi-objective optimization problem is of the form

$$\begin{aligned} &\min f(x) = [f_1(x), f_2(x), ..., f_k(x)]^\top. \\ &\text{subject to } x \in S \end{aligned} \tag{6}$$

where we have $k \geq 2$ potentially conflicting objective functions $f_i \colon \mathbb{R}^n \to \mathbb{R}$ that we wish to minimize simultaneously. The decision vector $x = (x_1, x_2, ..., x_n)^\top$ belongs to the nonempty feasible region $S \subset \mathbb{R}^n$. The vector function $f \colon \mathbb{R}^n \to \mathbb{R}^k$ is composed by k scalar objective functions. In MO-Op, the sets $\mathbb{R}^n$ and $\mathbb{R}^k$ are respectively known as decision variable space and objective space; they can also be defined as subsets of $\mathbb{R}^n$ and $\mathbb{R}^k$. The decision vector is regarded as optimal in MO-Op if they satisfy (6).

1. *Pareto Optimality*

In single-objective optimization the relation "less than or equal ($\leq$)" is used to compare the values of the scalar objective functions. In contrast, in multi-objective problems, there is no canonical order on $\mathbb{R}^k$ and thus we need other definitions of order to compare vectors in $\mathbb{R}^k$.

In MO-Op, the Pareto dominance relation is usually adopted, originally proposed by Edgeworth in [17] but generalized by French-Italian economist, Vilfredo Pareto in [18].

Pareto Dominance Relation: We say that a vector z^1 Pareto dominates vector z^2, denoted by $z^1 \prec z^2$, if and only if:

$$\begin{gathered} \forall i \in \{1,...,k\}: z_i^1 \leq z_i^2 \\ \text{and} \\ \exists i \in \{1,...,k\}: z_i^1 < z_i^2 \end{gathered} \tag{7}$$

Pareto Optimality: In MO-Op, a decision vector $x^* \in S$ is called Pareto Optimal if there does not exist another $x \in S$ such that $f(x) \prec f(x^*)$.

Weak Pareto Optimality: A solution $x^* \in S$, is weakly Pareto optimal if there does not exist another solution $x \in S$ such that $f(x) < f(x^*)$ for all $i = 1,...,k$..

Pareto Optimal Set: The Pareto optimal set, P^* is defined as:

$$P^* = \{x \in S \mid \nexists y \in S: f(y) \preceq f(x)\}. \tag{8}$$

Pareto Front: For a Pareto optimal set, P^*, the Pareto front, PF^*, is defined as:

$$PF^* = \{f(x) = [f_1(x), f_2(x), ..., f_k(x)]^\top \mid x \in P^*\}. \tag{9}$$

Figure 2 below illustrates the concept of Pareto optimal set and its image in the objective space, the Pareto front. Darker points denote Pareto optimal vectors. In variable space, these vectors are referred to as Pareto optimal decision vectors, while in objective space, they are called Pareto optimal objective vectors. As it is shown in Figure 2, the Pareto front is only composed by non-dominated vectors.

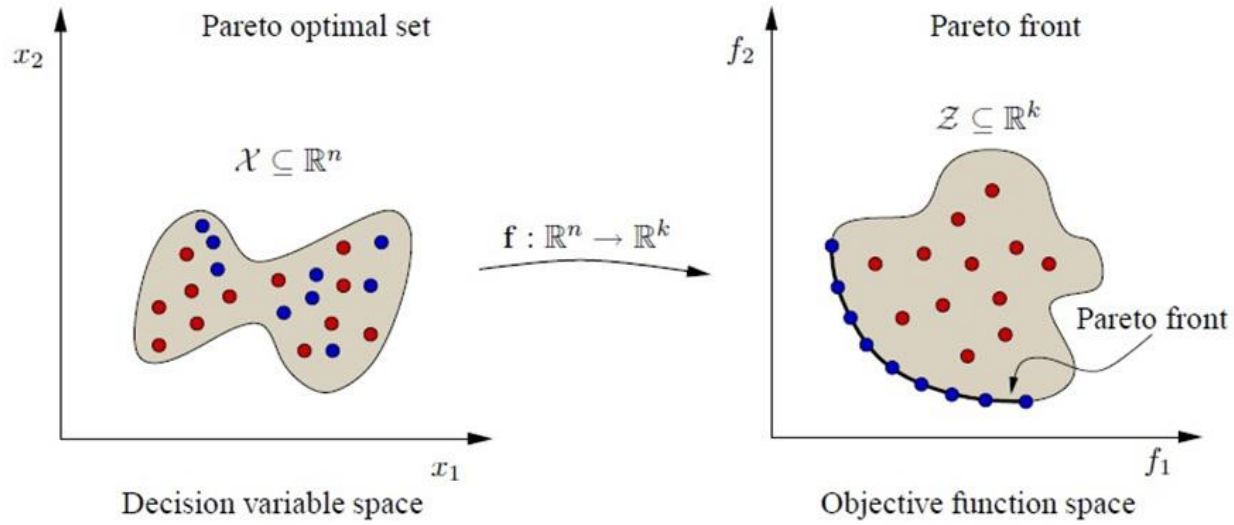


Figure 2. The Pareto optimal set and its image, Pareto front.

2. *Mathematical Programming Techniques*

The mathematical programming techniques are classified regarding how and when to incorporate preferences from the decision maker into the research process. Since mathematically the Pareto optimal decisions are equivalent, a decision maker is required to provide ordering. The decision maker is the person who shows his/her preferences among the objective and defines a tradeoff to be applied. A very important issue is the moment at which the decision maker is required to provide preference information. There are three ways of doing this:[19]

1) Prior to the search (a priori approaches), 2) During the search (interactive approaches), 3) After the search (a posteriori approaches).

Now, we explain some of the well-known methods in each category.

1) A priori Methods

Goal Programming: Charnes and Cooper[20] are credited with the development of the goal programming method for a linear model, and played a key role in applying it to industrial problems. In this method, the decision maker has to assign targets or goals that wishes to achieve for each objective. These values are incorporated into the problem as additional constraints. The objective function then tries to minimize the absolute deviations from the targets to the objectives. The simplest form of this method may be formulated as follows:

$$\begin{gathered}\min \sum_{i=1}^{k} |f_i(x) - T_i|. \\ \text{subject to } x \in S\end{gathered} \tag{10}$$

where T_i denotes the target or goal set by the decision maker for the ith objective function. A more general formulation of the goal programming objective function is a weighted sum of the pth power of the deviation $|f_i(x) - T_i|$. Such a formulation has been called generalized goal programming.
In the previous equation, the objective function is nonlinear and the simplex method can be applied only after transforming this equation into a linear form, thus reducing goal programming to a special type of linear programming. In this transformation, new variables δ_i^+ and δ_i^- are defined such that:

$$\begin{aligned}\delta_i^+ &= \tfrac{1}{2}\{|f_i(x) - T_i| + |f_i(x - T_i)|\}. \\ \delta_i^- &= \tfrac{1}{2}\{|f_i(x) - T_i| - |f_i(x - T_i)|\}.\end{aligned} \tag{11}$$

Which are underachievement and overachievement variables. So, the resulting equivalent linear formulation may be found:

$$\begin{gathered}\min \sum_{i=1}^{k} (\delta_i^+ + \delta_i^-). \\ \text{subject to } f(x) - \delta_i^+ + \delta_i^- = T_i, i = 1, ..., k \\ \delta_i^+, \delta_i^- \geq 0, i = 1, ..., k \\ x \in S\end{gathered} \tag{12}$$

Since it is not possible to have both under and overachievements of the goal simultaneously, we have:

$$\delta_i^+ \cdot \delta_i^- = 0.$$

Lexicographic Method: In this method, the objectives are ranked in order of importance by the decision maker (from best to worst). The optimal value $f_i^*(i = 1, ..., k)$ is then obtained by minimizing the objective functions sequentially, starting with the most important one and proceeding according to the order of importance of the objectives. Additionally, the optimal value found of each objective is added as a constraint for subsequent optimizations. This way, the optimal value of the most important objectives is preserved. Only in the case of several optimal solutions in the single optimization of the current objective, the rest of the objectives will be considered. Therefore, in the worst case, we have to carry out k single objective optimizations.

2) Interactive Methods

Light Beam Search: The Light Beam Search (LBS) method proposed by Jaszkiewicz and Slowinski,[21] is an iterative method which combines the reference point idea and tools of Multi-Attribute Decision Analysis (MADA). At each iteration, a finite sample of non-dominated points is generated. The sample is composed of a current point called middle point, which is obtained in previous iteration, and J non-dominated points from its neighborhood. A local preference model in the form of an outranking relation S is used to define the neighborhood of the middle point. It is said that a outranks b (aSb), if a is considered to be at least as good as b. The outranking relations is defined by decision maker, which specify three preference thresholds for each objective. They are indifference threshold, preference threshold and veto threshold. The decision maker has the possibility to scan the inner area of the neighborhood along the objective function trajectories between any two characteristic neighbors or between a characteristic neighbor and the middle point.

The next three methods relax the problem in different ways so that it can be posed as a single objective optimization problem.

Chebyshev Method: Chebyshev method proposed in [22], is an iterative method that requires user inputs. This method is based on the minimization of a function value, assuming that the global ideal objective vector (Utopian vector, the vector that optimizes all the objective function which often does not exist) is known. The metric to be used for measuring the distances to a utopian objective vector is the weighted Chebyshev metric. Thus, the multi-objective

optimization problem is transformed into a single-objective optimization problem, defined by:

$$\begin{gathered}\min \max[\omega_i(f_i(x)-z_i^*)], i=1,...,k. \\ \text{subject to } x \in S\end{gathered} \tag{13}$$

where $W=\{\omega \in \mathbb{R}^k \mid 0<\omega_i<1, \sum_{i=1}^{k} \omega_i = 1$ and z^* is the utopian objective vector.

3) A posteriori Methods

Linear combination of weights: In this method, the general idea is to associate each objective function with a weighting coefficient and minimize the weighted sum of the objectives. In this way, the multi-objective problem is transformed into a single objective problem.

$$\begin{gathered}\min \sum_{i=1}^{k} \omega_i f_i(x). \\ \text{subject to } x \in S\end{gathered} \tag{14}$$

where $\omega_i \geq 0$ and is strictly positive for at least one objective, such that $\sum_{i=1}^{k} \omega_i = 1$. Linear Quadratic Regulators utilize this technique for optimizing over both the control input and the regulation error.

ε-constraint method: In this method, one of the objective functions is selected based on a priority to be optimized and the other objectives are posed as the constraints to the single objective optimization problem. The problem is posed as

$$\begin{gathered}\min f_l(x). \\ \text{subject to } f_j(x) \leq \varepsilon_j \text{ for all } j=1,...,k,\ j \neq l, x \in S\end{gathered} \tag{15}$$

where ε_j are upper bounds for the objectives. The idea is to relax the MO-Op problem by posing it as a constrained optimization problem.

The different methods discussed above, each have their benefits and disadvantages. Typically, the weighting method,[23] and the ε-constraint[24] are more suitable in online autonomous decision making. However, it is not always possible to find a linear combination of weights for obtaining non-conservative MO-Op solutions. Here we use the ε-constraint method since the constrained optimization problem is better suited for online flight control problems. This is because online flight control problems are often designed to accommodate other constraints most of the time, such as maximum allowable inputs or state constraints. Hence available constrained optimization routines such as quadratic optimization can be utilized. Also, with this method we avoid finding the corner solutions (extreme solutions) to the problem and this makes the algorithm computationally efficient. In ε-constraint method, there is no need to scale the objectives to obtain the final solution and finally our MPC structure is simpler to implement and consistent with this solution method.

B. MO-Op Model Predictive Control (MPC)

The main idea in our architecture is to pose the MO-Op problem as a single objective constrained optimization problem (ε-constraint method) which can then be solved online using well-studied techniques such as MPC. In this section a brief overview of MPC for constrained discrete systems is presented and it is discussed how MPC can be used to obtain optimal solutions for the reference command shaping. The discretized version of the reference model dynamics is formulated:

$$\begin{aligned}x_{rm}(k+1) &= A_{rm}x_{rm}(k)+B_{rm}u_{rm}(k), \\ y(k) &= C_{rm}x_{rm}(k).\end{aligned} \tag{16}$$

by taking a difference operation on both sides we have:

$$\begin{aligned}\Delta x_{rm}(k+1) &= A_{rm}\Delta x_{rm}(k)+B_{rm}\Delta u_{rm}(k), \\ \Delta y(k) &= C_{rm}\Delta x_{rm}(k).\end{aligned} \tag{17}$$

Defining $\bar{x}_{rm}(k)=[\Delta x_{rm}^T(k) \quad y(k)]^T$ results in the following state-space model:

$$\begin{aligned}\bar{x}_{rm}(k+1) &= \bar{A}_{rm}\bar{x}_{rm}(k)+\bar{B}_{rm}\Delta u_{rm}(k), \\ y(k) &= \bar{C}_{rm}\bar{x}_{rm}(k).\end{aligned} \tag{18}$$

where the triplet $(\bar{A}_{rm}, \bar{B}_{rm}, \bar{C}_{rm})$ which is called the augmented model, is as follows:

$$\bar{A}_{rm} = \begin{bmatrix} A_{rm} & o_{rm}^T \\ C_{rm}A_{rm} & 1 \end{bmatrix} \bar{B}_{rm} = \begin{bmatrix} B_{rm} \\ C_{rm}B_{rm} \end{bmatrix} \bar{C}_{rm} = \begin{bmatrix} o_{rm} & 1 \end{bmatrix}$$

Upon formulation of the mathematical model, the next step in design of a predictive controller is to calculate the predicted plant output with the future control signal as the adjustable variables.[25] Assuming that at the sampling instant $k_i > 0$, the future control trajectory is denoted by:

$$\Delta u(k_i), \Delta u(k_i+1), \ldots, \Delta u(k_i+N_c-1).$$

Also, the future state variables are,

$$x(k_i+1 \mid k_i), x(k_i+2 \mid k_i), \ldots, x(k_i+N_p \mid k_i).$$

where N_p and N_c are prediction and control horizon, respectively. Based on the state-space model (18), the future state and output variables are calculated sequentially using the set of future control parameters and form the compact matrix equation (19) by defining:

$$Y = \begin{bmatrix} y(k_i+1 \mid k_i) \\ y(k_i+2 \mid k_i) \\ \vdots \\ y(k_i+N_p \mid k_i) \end{bmatrix} \Delta U = \begin{bmatrix} \Delta u((k_i) \\ \Delta u((k_i+1) \\ \vdots \\ \Delta u((k_i+N_c-1) \end{bmatrix}$$

$$Y = Fx(k_i) + \phi \Delta U. \tag{19}$$

where

$$F = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{N_p} \end{bmatrix} \phi = \begin{bmatrix} CB & 0 & 0 & \ldots & 0 \\ CAB & CB & 0 & \ldots & 0 \\ CA^2B & CAB & CB & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & \ldots & CA^{N_p-N_c}B \end{bmatrix}$$

For a given set-point signal $r(k_i)$ at sample time k_i, the optimization objective is to find the best control parameter vector ΔU such that an error function between the set-point and the predicted output is minimized.

Assuming that the data vector which contains the set-point information is $R_s^T = \begin{bmatrix} 1 & 1 & \ldots & 1 \end{bmatrix} r(k_i)$, the cost function J that reflects the control objective is,

$$J = (R_s - Y)^T Q(R_s - Y) + \Delta U^T \bar{R} \Delta U. \tag{20}$$

Here, Q denotes a positive definite diagonal matrix and $\bar{R} = r_w I_{Nc}$ which r_w is a tuning parameter for desired closed-loop performance. If we insert equation (19) into (20), we could form the MPC problem as minimizing the cost function,

$$\begin{aligned} J = {} & (R_s - Fx(k_i))^T Q(R_s - Fx(k_i)) \\ & - 2\Delta U^T \phi^T Q(R_s - Fx(k_i)) \\ & + \Delta U^T (\phi^T Q \phi + \bar{R}) \Delta U. \end{aligned} \tag{21}$$

which is subject to constraints on the incremental control ΔU and the output Y for the first objective, tracking error and on the system state for the second objective, oscillation (energy in the system), that are collected together in equation (22) as a matrix compact form.

$$\begin{bmatrix} -C_2 \\ C_2 \\ -\phi \\ \phi \\ C_2 \end{bmatrix} \Delta U \leq \begin{bmatrix} -U_{min} + C_1 u(k_i-1) \\ U_{max} - C_1 u(k_i-1) \\ -Y_{min} + Fx(k_i) \\ Y_{max} - Fx(k_i) \\ -C_1(\dot{x}_{rm_2} + \omega_n^2 x_{rm_1} + 2\zeta\omega_n x_{rm_2} - \Delta_{ub}) \end{bmatrix} \tag{22}$$

where Δ_{ub} is an upper bound on the unknown terms of our system dynamics. In the presented MPC architecture, constraints cannot be placed on the un-measured state. However, this is typically not an issue, since these constraints can be transformed into constraints on the control input. For example, the upper limit on the oscillation in the system results in an upper bound for the system states. Hence, using the state equations, a bound on the control inputs can be defined as shown above.

C. Model Reference Adaptive Control (MRAC)

In this section we briefly review MRAC.[12,26] We begin first by describing model reference control architecture for systems without any uncertainty. Let $x(t) \in R^n$ be the state vector, let $u(t) \in R^m$ denote the control input and consider the following linear system without uncertainty.

$$\dot{x}(t) = Ax(t) + Bu(t). \tag{23}$$

where $A \in R^n \times n, B \in R^n \times m$. We assume that the pair (A,B) is controllable and that B has full column rank. We assume $u(t)$ is restricted to the class of admissible control inputs consisting of measurable functions and $x(t)$ is available for full state feedback.

A designer chosen linear reference model is used to characterize the desired closed-loop response of the system,

$$\dot{x}_{rm}(t) = A_{rm}x_{rm}(t) + B_{rm}r(t). \tag{24}$$

The reference command $r \in R^m$ is assumed to be bounded and piecewise continuous, the matrix $A_{rm} \in R^{n\times n}$ is chosen to be Hurwitz and $B_{rm} \in R^{n\times m}$ is such that steady state accuracy is ensured. An adaptive control law including a linear feedback part $u_{fb}(t) = K_m^T x(t)$ with $K_m \in R^{n\times m}$, a linear feedforward part $u_{ff}(t) = K_b^T r(t)$ with $K_b \in R^{1\times m}$ is proposed to have the following form:

$$u(t) = u_{ff}(t) + u_{fb}(t). \tag{25}$$

Substituting (25) into (23) we have:

$$\dot{x}(t) = (A + BK_m^T)x(t) + BK_b^T r(t). \tag{26}$$

The design objective is to have (26) behave as the reference model in (24). To that effect, we introduce the following model matching conditions,

$$\begin{aligned} A + BK_m^T &= A_{rm}, \\ BK_b^T &= B_{rm}. \end{aligned} \tag{27}$$

Defining the tracking error to be $e(t) = x(t) - x_{rm}(t)$ yields,

$$\dot{e}(t) = A_{rm}e(t). \tag{28}$$

which shows that the tracking error converges to zero exponentially fast since A_{rm} is Hurwitz. It follows from Lyapunov theory that there exists a unique positive definite matrix $P \in R^{n\times n}$ satisfying the Lyapunov equation,

$$A_{rm}^T P + PA_{rm} + Q = 0. \tag{29}$$

for any positive definite matrix $Q \in R^{n\times n}$.

1. MRAC with uncertain nonlinearity

Here we discuss how the effect of an additive matched uncertainty can be negated using MRAC. Suppose that we have an uncertain nonlinear model with a matched modeling uncertainty Δ. Then equation (23) will be,

$$\dot{x}(t) = Ax(t) + B(u + \Delta)(t). \tag{30}$$

Consider of an adaptive term u_{ad} which needs to be designed to cancel out $\Delta(x)$. So, we modify our control law in equation (25), as

$$u(t) = u_{ff}(t) + u_{fb}(t) - u_{ad}(t). \tag{31}$$

With (31), (30) and (24), the same as what we did in (26), the tracking error dynamics would be:

$$\dot{e} = A_{rm}e + B(\Delta - u_{ad}). \tag{32}$$

Let Γ_w denote positive definite learning rate and consider the well-known gradient based adaptation law $\dot{W}(t) = -\Gamma_w w(t) e^T(t) PB$ that minimizes a cost on the instantaneous tracking error e guarantees that the tracking error is uniformly bounded for the adaptive controller framework described above. However, this adaptive law guarantees that the parameters (W) stay bounded within a neighborhood of the ideal parameters (W^*) only if $w(t)$ is persistently excited.[27] Chowdhary et al. in [28] used concurrent learning to overcome this problem. Furthermore, in [16] it was shown that when Gaussian Processes are used as adaptive elements, the architecture becomes capable of adapting to a wide class of uncertainties without having to pre-specify the structure of the adaptive elements. Here, we take advantage of GPs to learn the uncertainty in the model by modeling the adaptive element as the mean of an online learned GP.

D. Gaussian Process MRAC

In the presented architecture, the stochastic uncertainty Δ_i in (1) is modeled as a Gaussian process (GP) [see (2)]. A GP is a distribution over functions and a sample drawn from a GP will be a function completely characterized by the mean and variance of the GP.[29] In GPs, the function is modeled as a linear combination of covariance kernel functions.[30] A common choice for defining the covariance K is the squared exponential function:

$$k(x,y) = exp(-\frac{\| x-y \|^2}{2\sigma^2}). \tag{33}$$

although the presented method can accommodate other kernels. The kernel function generates a mapping ψ to an infinite dimensional Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$ such that k is an inner product defined by $k(x,y) = \langle \psi(x), \psi(y) \rangle_H$. In this RKHS, the mean m_i of the data generating stochastic process Δ_i is a linear combination of nonlinear bases:

$$m(x) = \sum_{j=1}^{\infty} \alpha_j k(x, x_j). \tag{34}$$

where $\alpha_i \in \mathcal{H}$ is a countably infinite dimensional vector, $k \in \mathcal{H}$ are the kernel functions evaluated over data points x_j vectors in the dual space. This rather general model can be learned directly from data, let $\mathcal{D} = \{(X_1, y_1)(X_2, y_2), \ldots, (X_n, y_n)\}$ be a set of state measurements sampled from an environment. Then assuming Normal distribution as the prior on the data, the GP posterior mean and variance of the GP given the measurements are estimated as:

$$\begin{bmatrix} y_t \\ y_{t+1} \end{bmatrix} \sim \mathcal{N}(0, \begin{bmatrix} K(\mathcal{D},\mathcal{D}) + \omega^2 I & k(\mathcal{D},x) \\ k(\mathcal{D},x)^\top & k(x,x) \end{bmatrix}), \tag{35}$$

The posterior distribution is computed by conditioning the joint distribution over the new measurement y_{t+1} to obtain,

$$p(y_{t+1} \mid y_t, x_t, \mathcal{D}) \sim \mathcal{N}(\hat{m}(x_{t+1}), \hat{\Sigma}(x_{t+1})), \tag{36}$$

where,

$$\begin{aligned} \hat{m}(x_{t+1}) &= \alpha^\top k(x_{t+1}, \mathcal{D}), \\ \hat{\Sigma}(x_{t+1}) &= k(x_{t+1}, x_{t+1}) - k(\mathcal{D}, x_{t+1})^\top [K(\mathcal{D},\mathcal{D}) + \omega^2 I]^{-1} k(\mathcal{D}, x_{t+1}). \end{aligned} \tag{37}$$

are the predictive mean and covariance respectively and $\alpha = K(\mathcal{D},\mathcal{D}) + \omega^2 I]^{-1} \bar{y}$. Hence, the mean is directly estimated from the set of available data. In essence, at any given point in time, an instantiation of the GP can be thought of as a Gaussian Radial Bases Function Network (RBFN). However, the main strength of the GP is that it does need to assume an a-priori allocation of RBF centers.

However, the main disadvantage of using the traditional GP regression techniques is that the covariance matrix increases in size as the size of $\mathcal{D}$ increases. In online applications, this can quickly become intractable as computing the inverse in (37) can become computationally intractable. It was shown in [16] that this problem can be alleviated using online sparsification techniques, and in particular Csato and Opper's budgeted online Sparse Gaussian Process regression technique[31] which only includes valuable data points in an active *Basis Vector set* ($\mathcal{BV}$). When new data is observed, the sparsification algorithm computes how well the new data point can be approximated by the existing basis vectors using a comparative test called the *kernel linear independence test* which $\forall \psi \in \mathcal{BV}$ is defined as:

$$\gamma_{t+1} = \| \sum_{i=1}^{t} \alpha_i \psi(x_i) - \psi(x_{t+1}) \|_{\mathcal{H}} . \tag{38}$$

The γ_{t+1} gives the residual distance between $\psi(x_i)$ and the GP generated by elements in $\mathcal{BV}$. An existing element ψ_m in the basis vector set which minimizes $D(\mathcal{GP} \parallel \mathcal{BV}) - D(\mathcal{GP} \parallel \mathcal{BV} \backslash \{\psi_m\})$ is removed and the new sample is added to the set. Given the basis vector set, the approximate mean and variance can be written from (37) as:

$$\begin{aligned} \hat{m}(x_{t+1}) &= \alpha^\top k(x_{t+1}, \mathcal{BV}), \\ \hat{\Sigma}(x_{t+1}) &= k(x_{t+1}, x_{t+1}) - k(\mathcal{BV}, x_{t+1})^\top [K(\mathcal{BV}, \mathcal{BV}) + \omega^2 I]^{-1} k(\mathcal{BV}, x_{t+1}). \end{aligned} \tag{39}$$

The reader is referred to [16] and [31] for details of the algorithmic implementation.

To achieve the tracking objective, the adaptive element is set to the online learned mean of the stochastic process Δ_i: $u_{ad}(x) = \hat{m}_i(x)$ at any state $x(t)$ from (5) to ensure that the tracking error goes to zero asymptotically.

$$\begin{aligned} u(t) &= u_{ff}(t) + u_{fb}(t) - u_{ad}(t), \\ u_{ad} &= \hat{m}(x). \end{aligned} \tag{40}$$

An online implementation of Gaussian Processes in MRAC is presented in [16].

IV. Results

In this section we evaluate the presented architecture (Figure 1) through simulation on a representative flight control problem. We consider the wing-rock dynamics problem.[32] However, we add to the problem an additive emulated rigid-flexible mode oscillatory term: $p^2 sin(p)$. The frequency of oscillation is modeled to increase with the roll rate p, while the amplitude is modeled to scale with the square of the roll rate. Consider the uncertain nonlinear dynamical model of wingrock dynamics with the additive term (1):

$$\begin{bmatrix} \dot{\phi} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ p \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \delta_a + \Delta + p^2 sin(p). \tag{41}$$

where $\delta_a = u$ in (41) is the aileron deflection, ϕ is the roll angle, and p is the roll rate. The generic nonlinear and time-varying uncertainty Δ takes on the following form for wingrock dynamics $\Delta = W_0 + W_1\phi + W_2 p + W_3|\phi|p + W_4|p|p + W_5\phi^3$. It is assumed that the oscillatory term has frequencies that are beyond the bandwidth of the actuator system. Therefore, it cannot directly be canceled using constrained control input. Therefore, finding a control input that minimizes the oscillation in the system forms the second objective in our MO-Op problem.

The natural frequency and the damping ratio of the chosen reference model are $\omega_{rm} = 2$ and $\zeta_{rm} = 0.5$, so that:

$$\begin{bmatrix} \dot{x}_{rm_1} \\ \dot{x}_{rm_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -4 & -2 \end{bmatrix} \begin{bmatrix} x_{rm_1} \\ x_{rm_2} \end{bmatrix} + \begin{bmatrix} 0 \\ 4 \end{bmatrix} u_{rm}. \tag{42}$$

The feedback and feedforward gains are $K_m = [-4 \quad -2]$, $K_b = 4$, respectively. The simulation runs for 50 seconds with a time step of 0.01 seconds.

First, we present the results for the GP-MRAC-MPC without the oscillation term. The only objective function here is the tracking error. The results are shown in Figure 3. It can be seen that the GP-MRAC-MPC architecture can achieve good tracking performance.

Now we present results with the oscillatory term added. The oscillation term cannot be directly canceled by GP-MRAC since the dynamics tend to get very aggressive as p increases. The left plot in Figure 4 shows the performance of the baseline GP-MRAC controller without MO-Op MPC. The results show that the oscillatory term results in suboptimal performance. To handle this issue in the MO-Op framework, the second objective is posed as a constraint in our MPC problem utilizing the ε-constraint MO-Op method. Since the energy in the system is a measure of oscillation, for minimizing the oscillation the root mean square of the roll rate can be computed and a control strategy can be sought that restricts the oscillation below a user specified upper bound. The bound on the root mean square of the roll rate results in an upper bound for roll rate itself. This bound was found by plotting the root mean square of the roll rate versus the optimization iteration index. The values greater than the upper bound give us the iteration indexes that roll rate exceeded the threshold. Utilizing this, an upper bound on the roll rate can be found. After evaluating that upper bound, the MPC problem can be posed in ε-constraint manner by utilizing that upper bound as a constraint on p as explained in section III.B. The results for the solution to the MO-Op MPC problem compared to the baseline with only tracking error as the objective are shown in Figure 4. It can be seen that the MO-Op technique outperforms the baseline since it takes into account the potential for exciting the oscillatory mode. It does so by constraining the control input to a region where the oscillations are expected to be below the user defined threshold.

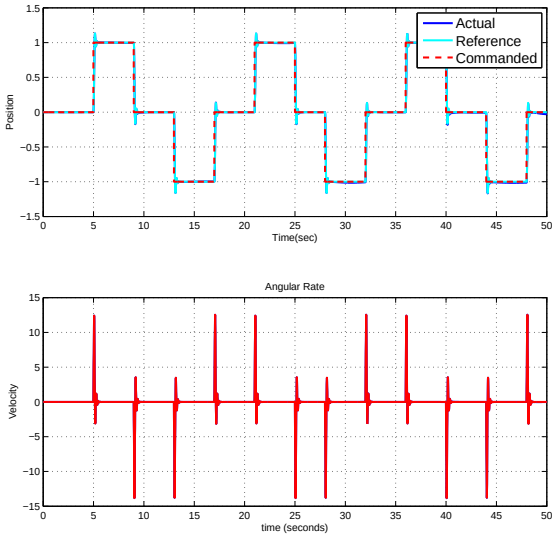
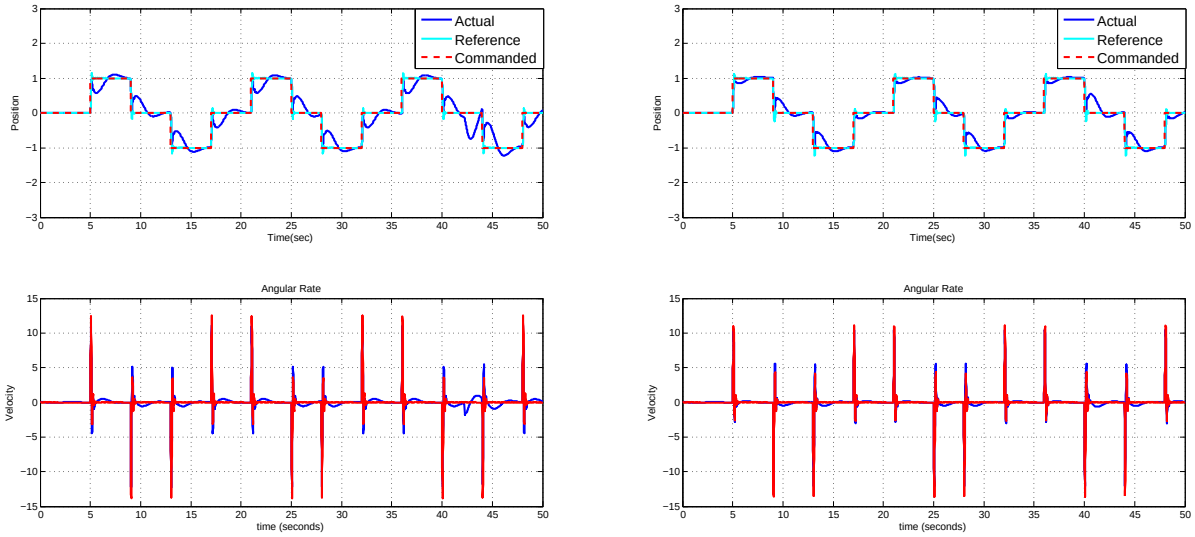


Figure 3. Position and velocity response for GP-MRAC-MPC architecture



(a) Performance without MO-Op

(b) Performance with MO-Op

Figure 4. Comparison in performance without and with optimizing over the oscillation.

V. Conclusion

We presented preliminary work towards a control architecture for multiple-objective optimization (MO-Op) flight control. Our goal was to demonstrate that utilizing a MO-Op framework, the problem of minimizing tracking error while simultaneously optimizing over other flight-performance relevant objectives can be addressed. Our architecture brings together learning based adaptive control with model predictive control. The learning is performed using Gaussian process adaptive elements, while the optimization is made realizable by relaxing some of the objectives by posing them as state constraints.The architecture was validated on a nonlinear roll dynamics system with wing rock effects and emulated state-dependent rigid-flexible interaction. Minimizing tracking error was one of the objectives, and minimizing the oscillation in the system to improve ride comfort was our second objective. The stability of key elements of the presented adaptive-optimal architecture is discussed in,[4] which makes this architecture applicable to many other multi-objective optimization problems.

Acknowledgments

This work was supported in part by University of Oklahoma NASA Cooperative Agreement No. NNX13AB21A. We thank Dr. Nhan Nguyen at NASA Ames for valuable discussions in formulating the MO-Op adaptive flight control problem.

References

[1]Nguyen, N. and Urnes, J., "Aeroelastic Modeling of Elastically Shaped Aircraft Concept via Wing Shaping Control for Drag Reduction," *AIAA Atmospheric Flight Mechanics Conference*, 2012, p. 1316.

[2]Branke, J., Deb, K., Miettinen, K., and Slowinski, R., *Multiobjective optimization: Interactive and evolutionary approaches*, Vol. 5252, Springer, 2008.

[3]Bertsekas, D. P., *Dynamic Programming and Optimal Control*, Vol. I–II, Athena Scientific, PO Box 391, Belmont, MA 02178, 2007.

[4]Mühlegg, M., Chowdhary, G., and Holzapfel, F., "Optimizing Reference Commands for Concurrent Learning Adaptive-Optimal Control of Uncertain Dynamical Systems," *GNC*, AIAA, 2013.

[5]Abdollahi, A., Allamaraju, R., and Chowdhary, G., "Adaptive-Optimal Control of Nonstationary Dynamical Systems," *Proceedings of the 2015 European Guidance, Navigation, and Control Conference*, 2015, accepted.

[6]Lee, J. H. and Cooley, B., "Recent advances in model predictive control and other related areas," *AIChE Symposium Series*, Vol. 93, New York, NY: American Institute of Chemical Engineers, 1971-c2002., 1997, pp. 201–216.

[7]Qin, S. J. and Badgwell, T. A., "An overview of industrial model predictive control technology," *AIChE Symposium Series*, Vol. 93, New York, NY: American Institute of Chemical Engineers, 1971-c2002., 1997, pp. 232–256.

[8]Bouffard, P., Aswani, A., and Tomlin, C., "Learning-based model predictive control on a quadrotor: Onboard implementation and experimental results," *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, may 2012, p. 279284.

[9]Aswani, A., Gonzalez, H., Sastry, S. S., and Tomlin, C., "Provably safe and robust learning-based model predictive control," *Automatica*, Vol. 49, No. 5, 2013, pp. 1216–1226.

[10]Adetola, V., DeHaan, D., and Guay, M., "Adaptive model predictive control for constrained nonlinear systems," *Systems & Control Letters*, Vol. 58, No. 5, 2009, pp. 320–326.

[11]Fukushima, H., Kim, T.-H., and Sugie, T., "Adaptive model predictive control for a class of constrained linear systems based on the comparison model," *Automatica*, Vol. 43, No. 2, 2007, pp. 301–308.

[12]Chowdhary, G., Mühlegg, M., How, J. P., and Holzapfel, F., "Concurrent Learning Adaptive Model Predictive Control," *Advances in Aerospace Guidance, Navigation and Control*, edited by Q. Chu, B. Mulder, D. Choukroun, E.-J. Kampen, C. Visser, and G. Looye, Springer Berlin Heidelberg, 2013, pp. 29–47.

[13]Chowdhary, G., Muhlegg, M., How, J., and Holzapfel, F., "A Concurrent Learning Adaptive-Optimal Control Architecture for Nonlinear Systems," *CDC*, IEEE, 2013.

[14]Pareto, V., "The New Theories of Economics," *Journal of Political Economy*, Vol. 5, No. 4, Sep. 1897, pp. 485502.

[15]Chowdhary, G., Kingravi, H. A., How, J. P., and Vela, P. A., "Bayesian nonparametric adaptive control of time-varying systems using Gaussian processes," *American Control Conference (ACC), 2013*, IEEE, 2013, pp. 2655–2661.

[16]Chowdhary, G., Kingravi, H., How, J., and Vela, P., "Bayesian Nonparametric Adaptive Control Using Gaussian Processes," *Neural Networks and Learning Systems, IEEE Transactions on*, Vol. PP, No. 99, 2014, pp. 1–1.

[17]Edgeworth, F. Y., *Mathematical psychics: An essay on the application of mathematics to the moral sciences*, C. Keagann Paul, 1881.

[18]Persky, J., "Retrospectives: Pareto's law," *The Journal of Economic Perspectives*, 1992, pp. 181–192.

[19]Van Veldhuizen, D. A., "Multiobjective evolutionary algorithms: classifications, analyses, and new innovations," Tech. rep., DTIC Document, 1999.

[20]Charnes, A. and Cooper, W. W., "Nonlinear power of adjacent extreme point methods in linear programming," *Econometrica: journal of the Econometric Society*, 1957, pp. 132–153.

[21]Jaszkiewicz, A. and Słowiński, R., "The Light Beam Searchapproach–an overview of methodology applications," *European Journal of Operational Research*, Vol. 113, No. 2, 1999, pp. 300–314.

[22]Steuer, R. E. and Choo, E.-U., "An interactive weighted Tchebycheff procedure for multiple objective programming," *Mathematical programming*, Vol. 26, No. 3, 1983, pp. 326–344.

[23]Gass, S. and Saaty, T., "The computational algorithm for the parametric objective function," *Naval research logistics quarterly*, Vol. 2, No. 1-2, 1955, pp. 39–45.

[24]Haimes, Y. Y., Lasdon, L. S., and Wismer, D. A., "On a bicriterion formulation of the problems of integrated system identification and system optimization," *IEEE Transactions on Systems Man and Cybernetics*, , No. 1, 1971, pp. 296–297.

[25]Wang, L., *Model predictive control system design and implementation using MATLAB®*, springer, 2009.

[26]Narendra, K. S. and Annaswamy, A. M., *Stable Adaptive Systems*, Prentice-Hall, Englewood Cliffs, 1989.

[27]Sanner, R. M. and Slotine, J.-J., "Gaussian networks for direct adaptive control," *Neural Networks, IEEE Transactions on*, Vol. 3, No. 6, 1992, pp. 837–863.

[28]Chowdhary, G., Muhlegg, M., How, J. P., and Holzapfel, F., "A concurrent learning adaptive-optimal control architecture for nonlinear systems," *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, IEEE, 2013, pp. 868–873.

[29]Rasmussen, C. E., "Gaussian processes for machine learning," 2006.

[30]Rasmussen, C. and Williams, C., *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*, The MIT Press, 2005.

[31]Csató, L. and Opper, M., "Sparse on-line Gaussian processes," *Neural computation*, Vol. 14, No. 3, 2002, pp. 641–668.

[32]Monahemi, M. M. and Krstic, M., "Control of Wingrock Motion Using Adaptive Feedback Linearization," *Journal of Guidance Control and Dynamics*, Vol. 19, No. 4, August 1996, pp. 905–912.