# OPTIMAL DESIGN OF EVTOLS FOR URBAN MOBILITY USING ANALYTICAL TARGET CASCADING (ATC)

BY

PRAJWAL KUMAR CHINTHOJU

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Systems and Entrepreneurial Engineering
in the Graduate College of the
University of Illinois Urbana-Champaign, 2022

Urbana, Illinois

Adviser:

Associate Professor James T. Allison

## ABSTRACT

The aim of this thesis is to evaluate the capability of Analytical Target Cascading (ATC) for optimizing large scale engineering design optimization problems. Multidisciplinary Optimization (MDO) techniques have a subject of immense interest in the past few decades because of their ability to generate optimal designs for complex system of systems. Monolithic MDO methods are one category of MDO methods that pose the optimization problem as a single optimization problem. These methods are effective in generating an optimal design, but it can be challenging to implement these methods at an organizational level in engineering design practice. Meanwhile, distributed MDO methods that decompose the problem into different sub optimization problems offer additional modularity and flexibility required for implementation at an organizational level. Analytical target cascading (ATC) is one such distributed MDO method that most closely emulates a hierarchical organization structure. The problem of designing of an eVTOL (electric vertical takeoff and landing vehicle) was chosen to evaluate the performance of ATC because of its relevance to the future urban mobility solution space and because the design of eVTOLs involves optimization of several subsystems that are strongly coupled. The first part of this thesis focuses on the implementation and analysis of the results generated by ATC with respect to eVTOL design optimization. In the later part of this thesis, we set to lay out some of the general advantages of using distributed MDO methods over Monolithic MDO methods, as well as situations where the latter is more beneficial. In the final chapter, aspects involving computational expense and human factors effort are discussed to explore a set of cases where distributed MDO methods can be advantageous in engineering practice.

## ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor Professor James Allison for his guidance and support throughout my MS program. I would also like to thank Prof. Kai James for leading this project and providing valuable feedback. I would also like to thank my colleague Ghanendra Das, who performed a lot of the initial research and helped me during my ramp up phase. I would like to thank my parents for all their encouragement throughout my MS program.

I would lastly like to thank NASA for their constant feedback to guide this project in a meaningful direction and Center for Power optimization of Electro-Thermal Systems (POETS) and National Science Foundation (NSF) for funding this project.

This work was supported by the National Science Foundation Engineering Research Center for Power Optimization of Electro-Thermal systems (POETS) with cooperative agreements EEC-1449548. The opinions, findings, and conclusions or recommendations expressed are those of the author(s) and do not necessarily reflect the views of the National Science Foundation

**TABLE OF CONTENTS**

# CHAPTER 1: INTRODUCTION

## 1.1 Nomenclature and Acronyms

$m_x$ = Mass of component x
$r_{prop}$ = Radius of propeller
$c_{prop}$ = Chord length of propeller
mtow = Maximum take-off Weight
$b_{span}$ = Wingspan
V = Cruise Velocity
AR = Wing aspect ratio
$root_{AoA}$ = Angle of attack at root of the wing
$P_{hover}$ = Power required to be generated by each motor during the hover phase
$\eta_{motor}$ = Efficiency of motor
eVTOL = electric Vertical Take-Off and Landing aircraft
FOM = Figure of merit (rotor performance)
ATC = Analytical Target Cascading
ALC = Augmented Lagrangian Coordination
ALP = Augmented Lagrangian Penalty
SAND = Simultaneous Analysis and Design
AAO = All at Once
IDF = Individual Discipline Feasible
MDF = Multi-Disciplinary Feasible
ALC = Augmented Lagrangian Coordination

## 1.2 Multidisciplinary Optimization (MDO)

Over the past decade, there has been tremendous interest in design optimization of system of systems [1]. The term system of systems, as used here, is a reference to an interconnected system of components or subsystems which function in synergy. Each of these subsystems can have different analysis requirements and can require engineers to work on varied disciplines. For example, consider design of a commercial aircraft; an aircraft consists of many subsystems such as propeller, engine, wing, fuselage, and avionics. Some of the subsystems such as the wing require more than one disciplinary analysis (aerodynamics and structural analysis) while others might just require one disciplinary analysis. The subsystems are also interdependent. In this example, the engine and the propeller must be able to generate enough thrust to overcome the drag generated by wings and other structural elements of the aircraft (which is computed via aerodynamic analysis). Conventional optimization of this kind of system, where the subsystems are inter-dependent, can be challenging. Traditionally, the design of system of systems was carried out using individual optimization, i.e., each subsystem was optimized individually, neglecting the interfacing between these subsystems. The drawback here is that each subsystem is individually optimized but the overall system design might be suboptimal. Individually optimizing each subsystem ensures that the performance of each subsystem is minimized but disregards the tradeoffs required for optimizing the entire system.

Multidisciplinary Optimization (MDO) techniques aim to solve this problem by formulating the optimization problem in such a way that all the interactions between different subsystems are included in the formulation. MDO methods were originally developed as a means of including other disciplines in structural design optimization [2]. However, this kind of implementation focused on partitioning based on discipline rather than subsystems or components. But by replacing different disciplinary analyses as the decomposed elements by subsystems, this formulation could be used for optimization of complex systems. Thus, the focus of some researchers later shifted to optimization of complex systems such as commercial aircrafts, spacecrafts, business jets, etc.

### 1.3 EVTOL design optimization

In this thesis we chose the problem of designing an electric Vertical Take-Off and Landing Vehicle (eVTOL) as an engineering-scale design optimization problem for evaluating use of the ATC (Analytical Target Cascading) system optimization strategy. Specifically, we are looking at an eVTOL aircraft for urban mobility. There are many reasons for this choice. Firstly, design of eVTOLs is an active area of research in the future mobility space. eVTOLs, also known as air taxis, have received a lot of attention [3] for their ability to use vertical space for urban transportation, being agile, and enabling an environmentally friendly means of transportation. In the current day and age many aerospace startups are focusing on prototyping eVTOLs as a service-based model. Some of the startups that are actively prototyping eVTOL designs are Joby [4], Archer [5], and Wisk [6].

However, design of eVTOLs also comes with its own set of challenges. The main cause of concern is to deliver a useful range of operation and being able to carry a significant payload (8-10 passengers). This would mean heavy batteries and motors are required to achieve the required operating range and payload capacity, respectively. Another cause of concern is the noise generated during operation. Since eVTOLs are designed to fly at low altitudes the amount of noise generated should be minimal, and that means, in part, that the angular speed of the propellers should be restricted. A fully electric commercial airliner is not yet feasible because it has significantly higher payload and range requirements, which are not possible with the power density of motors and energy density of batteries that we have today. Consequently, many non-conventional designs of eVTOLs have evolved in the past few years including configurations such as tiltrotor, tilt wing, rotorcraft, etc. Most of the design generation is carried out using monolithic MDO methods on reduced order models to rapidly explore the design space to find optimal designs. However, challenge arises when the design of the system needs the involvement of much higher fidelity models with interactions between different subsystems. This is a characteristic of many complex engineering design problems and using a monolithic MDO problem formulation can be a bottleneck as its successful construction and application requires many interactions across an engineering team. Different engineering teams cannot work in isolation while using a monolithic implementation, and the design organization must arrive at a consensus on an appropriate MDO problem formulation for the entire complex system. Although the computational effort required for monolithic MDO studies is often lower than for distributed MDO methods, the human effort required to collectively create, apply, and refine the problem formulation may outweigh any computational benefit. By evaluating the feasibility of using a distributed MDO method for engineering optimization problems at such scales, we can analyze the proximity of distributed MDO solution to the monolithic MDO solution and gain insight into the potential reduction of overall human effort.

## 1.4 Thesis Outline

The first few chapters of this thesis focus on introducing the concept of MDO and its classification, and the later section focuses on the eVTOL design problem and citing the potential benefit of using MDO in terms of human effort reduction. The second chapter begins with an introduction to MDO methods, laying out different formulations of monolithic MDO methods, and then goes into detail about ATC. An example of solving the Sellar problem using

ATC is also included for reference. The third chapter introduces the eVTOL problem, motivation, and mission statement. It lays out the objectives and constraints of this optimization problem and explains some of the models used in this formulation. The fourth chapter covers the implementational aspects of ATC on the eVTOL problem, such as the overall ATC structure used, penalty functions, and choice of coupling and local variables. The fifth chapter goes into detail about the human effort involved in distributed MDO methods and the potential benefits of using MDO methods. The thesis finally ends with a conclusion on how this study can be useful for further research in this domain.

# CHAPTER 2: MDO AND ATC FORMULATION

## 2.1 History of MDO

The development of Multidisciplinary optimization (MDO) methods began more than 30 years ago [7]. Before its development, complex engineered systems were designed using system engineering frameworks that had significant limitations. The conventional systems engineering framework involved a lot of interactions among different teams and was highly inefficient, leading to longer development times and delays in projects. For example, Boeing's 787 was nearly two years late to market owing to scheduling issues, and it also came with a cost overrun of $10 billion [8]. Undoubtedly, engineering systems are becoming more and more complex with increased customer demands and governmental regulations. The transition to electric vehicles is a good example of this.

The focus of early MDO method development efforts was to address these challenges and produce a rigorous strategy for complex engineering design optimization. In particular, the focus was on ensuring that all the system couplings are accounted for, and, at the same time, we also converge to an optimal system design. This led to the development of many different MDO methods, each of which have their own strength and weakness.

## 2.2 Classification of MDO methods

Fortunately, there is a convenient way to classify all these methods. Each of these methods involve either solution of a single optimization problem or the solution of multiple interrelated optimization problems. MDO methods that use a single optimization function are called Monolithic MDO methods. The other class of methods is referred to as distributed MDO.

The use of monolithic formulations for eVTOL optimization will serve as a helpful benchmark for comparison to distributed MDO. Therefore, it is useful to gain a brief understanding of these methods. Four different monolithic MDO methods that exist in MDO literature will be elaborated in the next few sections.

### 2.3 All-at-Once (AAO) formulation

In the AAO formulation, single copies of design variables *x* for each subsystem are used in the problem formulation, and the disciplinary analysis for each subsystem is included as an optimization constraint (called the residual) within the optimization problem formulation. The formulation also accounts for analysis variables, *y*. Copies of appropriate analysis variables are used in distinct disciplinary analyses, and while these may differ from each other during the optimization process, at convergence the local copies of these variables must match. This requirement is enforced by the consistency constraint. The concept of residuals and consistency constraints will be used extensively in later chapters. The general form of AAO is as given below:

$$\min_{x,y^t,y,\bar{y}} f_0(x,y) + \sum_{i=1}^{N} f_i(x_0, x_i, y_i)$$

$$\text{subject to}: c_0(x,y) \geq 0$$

$$c_i(x_0, x_i, y_i) \geq 0 \text{ for } i = 1, \dots, N$$

$$c_i^c = y_i^t - y_i = 0 \text{ for } i = 1, \dots, N$$

$$R_i\left(x_0, x_i, y_{j \neq i}^t, \bar{y}_i, y_i\right) = 0 \text{ for } i = 1, \dots, N$$

**( 1 )**

From the above problem formulation, we can see that some equality constraints within this problem formulation can be easily eliminated by substitution to produce alternative mathematically equivalent formulations. Depending on which constraints are eliminated, we can derive the rest of the monolithic MDO formulations. Please note that the above objective function assumes a specific form involving a general function of all variables summed with functions of variable subsets. When such a function structure exists, some MDO methods can leverage it for computational benefit. MDO problems do not necessarily need to have this objective function structure.

### 2.4 Simultaneous Analysis and Design (SAND) formulation

One simple way of making the above problem formulation easier to solve is by eliminating the consistency constraints below

$$c_i^c = y_i^t - y_i = 0 \text{ for } i = 1, \dots, N$$

This can be achieved by using only one set of variables for all the analyses. This would yield the following general formulation:

$$\min_{x,y,\bar{y}} f_0(x, y)$$

$$\text{subject to: } c_0(x, y) \geq 0 \text{ for } i = 1, \ldots, N$$
$$c_i(x_0, x_i, y_i) \geq 0 \text{ for } i = 1, \ldots, N$$
$$R_i\left(x_0, x_i, y^t_{j\neq i}, \bar{y}_i, y_i\right) = 0 \text{ for } i = 1, \ldots, N$$

(2)

Although this formulation is more efficient than AAO, other issues remain, i.e., the problem formulation still includes all the state/analysis variables and disciplines and has a large design representation. This can lead to premature termination of the optimization solver or other numerical challenges in many cases.

## 2.5 Individual Discipline feasible (IDF)

The IDF formulation can be derived by eliminating the residual constraints:

$$R_i\left(x_0, x_i, y^t_{j\neq i}, \bar{y}_i, y_i\right) = 0 \text{ for } i = 1, \ldots, N$$

This can be achieved by including the disciplinary analysis as an implicit function within the problem formulation. The general formulation of IDF is given by:

$$\min_{x,y^t} f_0\left(x, y(x, y^t)\right)$$

$$\text{subject to: } c_0\left(x, y(x, y^t)\right) \geq 0$$
$$c_i\left(x_0, x_i, y_i\left(x_0, x_i, y^t_{j\neq i}\right)\right) \geq 0 \text{ for } i = 1, \ldots, N$$
$$c^c_i = y^t_i - y_i\left(x_0, x_i, y^t_{j\neq i}\right) = 0 \text{ for } i = 1, \ldots, N$$

(3)

Note that we are back to using local copies in this formulation. This is also very similar in its advantages with SAND, i.e., it reduces the problem size, but the problem formulation is still large and complex. Additionally, not every analysis function can be made implicit and therefore IDF is limited in its application.

### 2.6 Multidisciplinary Feasible (MDF) formulation

After understanding the above formulations, one obvious suggestion that can be made is to eliminate both the consistency and residual constraints. This is exactly the how MDF is defined. The general formulation is therefore:

$$\min_{x} f_0\big(x, y(x, y)\big)$$
$$\text{subject to: } c_0\big(x, y(x, y)\big) \geq 0$$
$$c_i\Big(x_0, x_i, y_i\big(x_0, x_i, y_{j\neq i}\big)\Big) \geq 0 \text{ for } i = 1, \ldots, N \tag{4}$$

This formulation reduces the problem formulation size significantly. It has the limitations of both IDF and SAND in that it can only be applied to problems where disciplinary analysis can be made implicit. While it is a low-dimension problem, it can exhibit its own set of numerical challenges, especially those associated with use of a nested solver to find consistent state variables and the complex interplay between this solver and the optimization algorithm.

### 2.7 Analytical Target Cascading (ATC)

Monolithic MDO methods are sufficient for many problems involving coupled analysis. However, in the case of complex engineering design, multiple engineering teams need to design different components or subsystems and each team has its own expertise. Monolithic MDO methods can be difficult to implement in such scenarios because of the number of interactions that are required to 1) collectively define the detailed monolithic MDO problem, and 2) to iteratively refine the problem formulation, its solution, and converge on an optimal system design.

The challenge of these interactions is precisely the problem that distributed MDO methods aim to solve. In distributed MDO methods, the entire problem formulation is decomposed into multiple optimization problems based on a partioning criterion, such as components or disciplines. Each of these subsystems interact with other subsystems or the main optimization function via consistency constraints on coupling variables.  This ensures that each engineering team can work in isolation while also ensuring that synergy between different subsystems/disciplines is not lost.

Analytical target cascading (ATC) is a distributed MDO formulation that propagates system targets through a multilevel structure where each subsystem has its own optimization formulation [9],[10]. ATC was originally developed as an approach to cascade targets through the hierarchical structure for an engineering design organization [12]. Although ATC need not necessarily assume a hierarchical structure, in this study we focus on its hierarchical formulation given by Tosserams et al. [11]. The primary reason for choosing a hierarchical formulation is that it closely emulates propagation of design targets within engineering design practice. The formulation of a subproblem used within ATC is given below.

$$\min_{\bar{\mathbf{x}}_{ij}} \quad \mathrm{f}_{ij}(\bar{\mathbf{x}}_{ij}) + \phi(\mathbf{t}_{ij} - \mathbf{r}_{ij}) + \sum_{k \in \mathcal{C}_{ij}} \phi(\mathbf{t}_{(i+1)k} - \mathbf{r}_{(i+1)k})$$

$$\text{subject to } \mathbf{g}_{ij}(\bar{\mathbf{x}}_{ij}) \leq \mathbf{0}$$

$$\mathbf{h}_{ij}(\bar{\mathbf{x}}_{ij}) = \mathbf{0}$$

$$\text{with } \mathbf{r}_{ij} = \mathbf{a}_{ij}\left(\mathbf{x}_{ij}, \mathbf{t}_{(i+1)k_1}, \dots, \mathbf{t}_{(i+1)k_{ij}}\right)$$

$$\bar{\mathbf{x}}_{ij} = \left[\mathbf{x}_{ij}, \mathbf{r}_{ij}, \mathbf{t}_{(i+1)k_1}, \dots, \mathbf{t}_{(i+1)k_{cij}}\right]$$

**( 5 )**

where $\bar{\mathbf{x}}_{ij}$ is the vector of the design variables for subsystem *j* at level *i*, $\mathrm{f}_{ij}$ is the local objective function at that subsystem, $\mathbf{t}_{ij}, \mathbf{r}_{ij}$ are the targets and responses between subsystem *j* and its parent subproblem, respectively, $\mathbf{t}_{(i+1)k}, \mathbf{r}_{(i+1)k}$ are the targets and responses of subsystem *j* with its child subproblem(s), and $\mathbf{g}_{ij}$ and $\mathbf{h}_{ij}$ are the local constraints of the subsystem *j* at level *i*.

One noticeable difference of the above formulation compared against other MDO formulations is that we use a penalty function to satisfy the consistency constraints. As with previous MDO formulations, they are not separately included as the constraints in the optimization solver. Instead, another coordination algorithm known as the Augmented Lagrangian Coordination is used to iteratively guide penalty functions such that consistency constraints are satisfied at the end of an ATC solution [13]. This coordination method makes use of quadratic and linear weights to adjust the penalty function weights between optimization executions. The iterative solution of ATC optimization subproblems terminates when the required consistency tolerance is achieved. The penalty function used in ALC is as shown below:

$$\phi(\mathbf{t}_{ij} - \mathbf{r}_{ij}) = \mathbf{v}_{ij}^{\mathrm{T}}(\mathbf{t}_{ij} - \mathbf{r}_{ij}) + \left\|\mathbf{w}_{ij} \circ (\mathbf{t}_{ij} - \mathbf{r}_{ij})\right\|_2^2$$

**( 6 )**

The weights **v** and **w** are updated between subproblem optimization solutions as follows:

$$\mathbf{v}^{k+1} = \mathbf{v}^{k} + 2\mathbf{w}^{k} \circ \mathbf{w}^{k} \circ \mathbf{q}^{k}$$

**( 7 )**

$$w_i^{k+1} = \begin{cases} w_i^k & \text{if } \left|q_i^k\right| \leq \gamma\left|q_i^{k-1}\right| \\ \beta w_i^k & \text{if } \left|q_i^k\right| > \gamma\left|q_i^{k-1}\right| \end{cases}$$

**( 8 )**

$q^k$ here is $(\mathbf{t}_{ij} - \mathbf{r}_{ij})$ at $k$th iteration ($k$th optimization subproblem solution) and the ∘ operator refers to the Hadamard product: an element-wise product of two same dimensioned matrices (that generates another matrix of the same dimension).

Gamma (γ) and beta (β) are two hyperparameters which can be tuned to improve ATC convergence properties. Gamma controls the threshold at which the weights need to be updated. For example, if the consistency constraints have already converged by a ratio of gamma in the previous iteration, then the quadratic weights need not be increased, and the update step is skipped. Note that the linear weights are adjusted whether consistency constraints have converged by a ratio of gamma or not. Hence, the quadratic weights are only increased if the constraint value is not converging as per expectations. Beta, on the other hand, controls the rate at which the quadratic weights grow with each iteration.

The ATC solution process starts by solving the system level optimization problem. The system level optimization problem usually includes the overall objective function and constraints at the system level. The resulting solution from the system level optimization produces the targets for all the subsystems in the next level. The algorithm then proceeds to solve the individual subsystem optimization for all the subsystems in the second level (this need not be sequential, parallel and other coordination strategies exist). The algorithm then continues to repeat the same for all the levels in the hierarchy. Once all the levels are addressed, the algorithm checks for consistency constraint tolerances and updates the weights if the tolerances are not met. After that, the cycle begins again with system level optimization. This cycle is repeated until the consistency constraint tolerances are met, where the algorithm stops, and the resulting solution is the optimal solution. Since the criteria for stopping the algorithm is based on consistency constraint tolerance, we need not be concerned that the resulting design variable copies at each subsystem are inconsistent. This algorithm is articulated as a simple flow chart in the figure below:

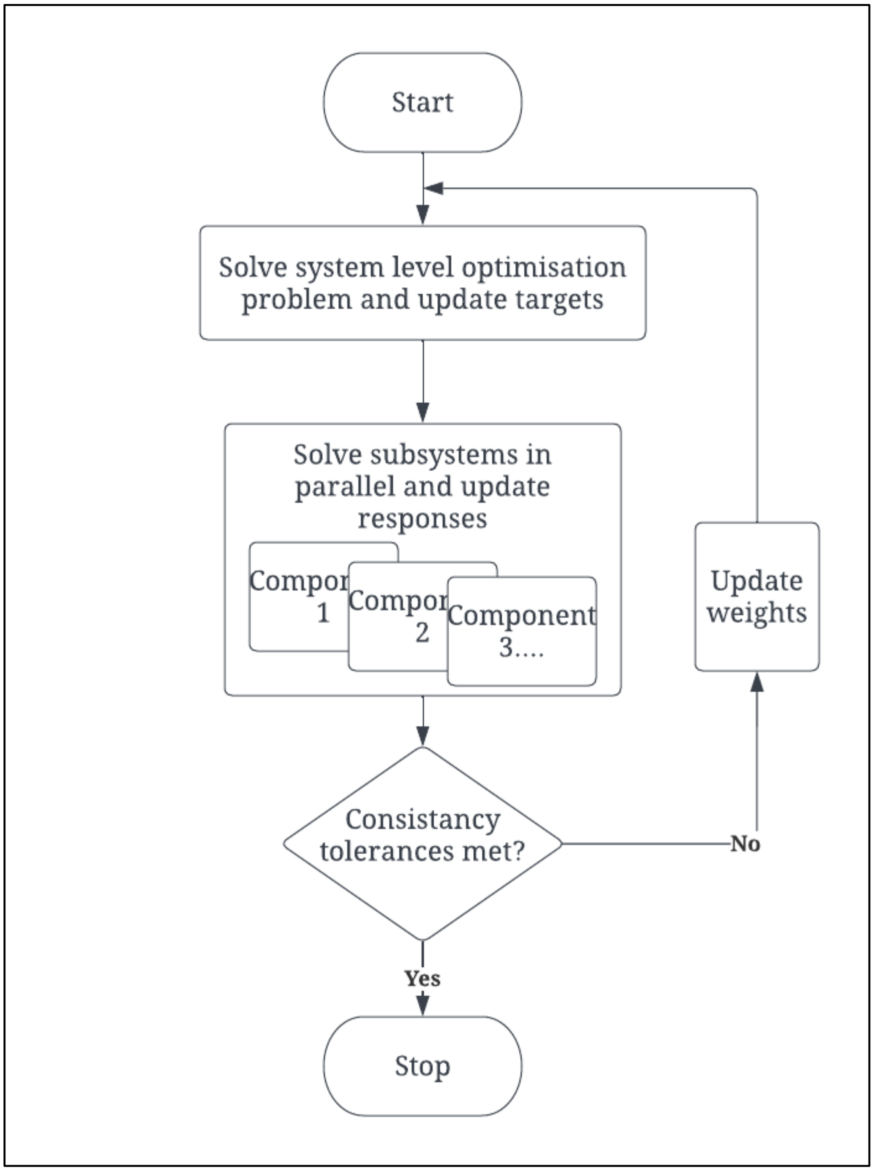



**Figure 2.1: Flow chart of ATC algorithm**

There are a few challenges in the implementation of ATC for engineering scale problems. From the flow chart representation above, one can spot that ATC may be computationally more expensive than monolithic options because it is a nested loop of optimization problems. The figure below also shows the comparison between monolithic architectures (MDF, SAND, IDF) and other distributed architectures (Collaborative Optimization, CO) [26]. Although performance analysis of ATC was not included in the study that produced the results in this figure was, ATC will also follow a similar trend given that it is also a distributed MDO methods. In the next section, the implementation of ATC on the Sellar problem is demonstrated.

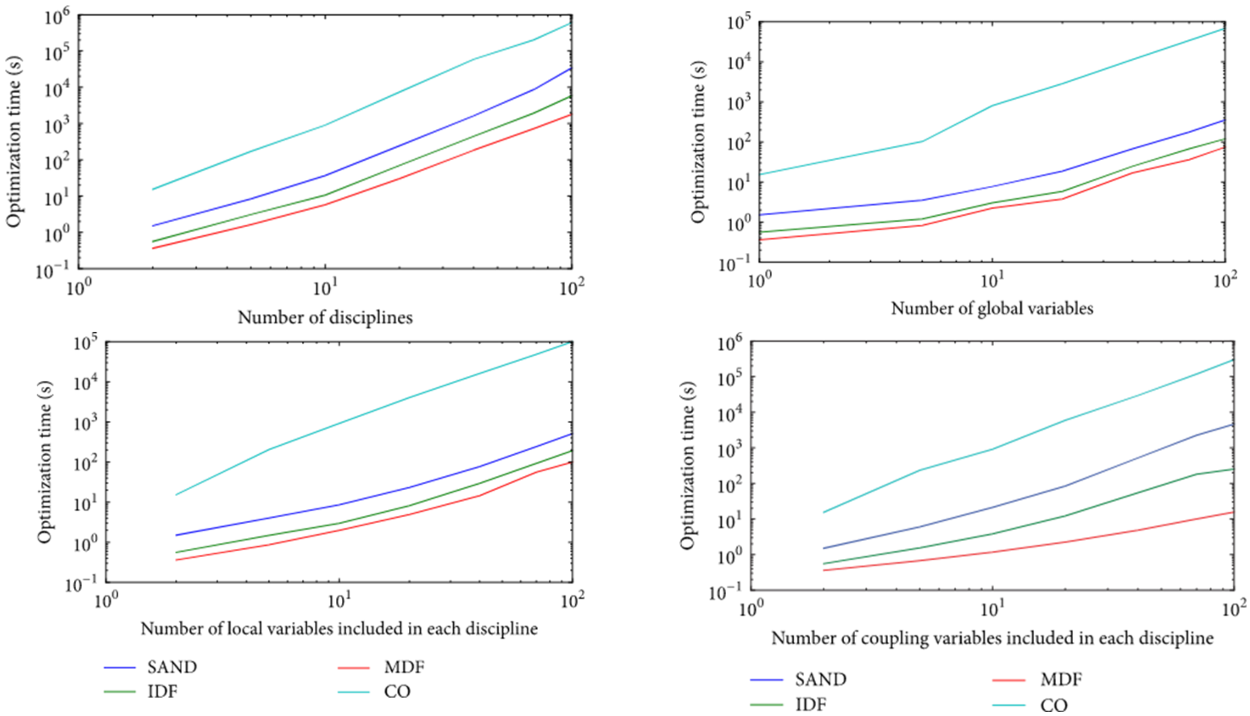



**Figure 2.2: Performance comparison of distributed vs monolithic MDO [26]**

## 2.8 Implementation of ATC on the Sellar test problem

The Sellar test problem is a coupled disciplinary problem that is designed to evaluate MDO formulations [14]. It is also included as an example problem in the OpenMDAO documentation. For this study, we compare the solutions of both the AAO and ATC solutions of the Sellar problem. The Sellar problem formulation is given as follows:

$$\min_{z_1,z_2,x_1} x_1^2 + z_2 + y_1 + e^{-y_2}$$
$$\text{subject to: } \frac{y_1}{3.16} - 1 \geq 0$$
$$1 - \frac{y_2}{24} \geq 0$$
$$-10 \leq z_1 \leq 10$$
$$0 \leq z_2 \leq 10$$

$$0 \leq x_1 \leq 10$$

$$\text{Disipline 1:} \quad y_1(z_1, z_2, x_1, y_2) = z_1^2 + x_1 + z_2 - 0.2y_2$$
$$\text{Discipline 2:} \quad y_2(z_1, z_2, y_1) = \sqrt{y_1} + z_1 + z_2$$

**( 9 )**

The design variables here are $z_1, z_2$ and $x_1$, and $y_1$ and $y_2$ are the coupled variables. The variables $z_1$ and $z_2$ are called global variables because they are included in all the disciplines. The variable $x_1$ is a local variable because it is used only in discipline 1. $y_1$ and $y_2$ are called coupling variables as they are the result of individual disciplinary analysis and are used in other disciplines.

The global optimum solution for this problem is also available in Ref. [14]. Below are the optimal values of the Sellar problem as given in Ref. [14].

**Table 2.1: Optimal values of Sellar problem**

| Design variable | Value |
|---|---|
| $\mathbf{x_1}$ | 1.977 |
| $\mathbf{z_1}$ | 0 |
| $\mathbf{z_2}$ | 0 |
| $\mathbf{y_1}$ | 3.16 |
| $\mathbf{y_2}$ | 3.755278 |
| Min. objective | 3.18339 |

With this understanding of the sellar problem, a preliminary optimization of the sellar problem with SAND formulation is performed. The SAND formulation of this problem is as follows:

$$\min_{z_1, z_2, x_1} x_1^2 + z_2 + y_1 + e^{-y_2}$$
$$\text{subject to:}$$
$$y_1(z_1, z_2, x_1, y_2) = z_1^2 + x_1 + z_2 - 0.2y_2$$
$$y_2(z_1, z_2, y_1) = \sqrt{y_1} + z_1 + z_2$$
$$\frac{y_1}{3.16} - 1 \geq 0$$

$$1 - \frac{y_2}{24} \geq 0$$
$$-10 \leq z_1 \leq 10$$
$$0 \leq z_2 \leq 10$$
$$0 \leq x_1 \leq 10$$

**( 10 )**

In the above formulation we can see that both the disciplinary analyses have been included as constraints and only one set of design variables is used in the entire analysis, avoiding the use of local copies of design variables. This is exactly the approach we discussed in the SAND formulation from Section 2.4. The resulting optimum solution is as follows (Note: the `fmincon` function from MATLAB was used for optimization).

**Table 2.2: Optimal values of Sellar problem using SAND formulation**

| Design variable | Value |
|---|---|
| $\mathbf{x_1}$ | 0 |
| $\mathbf{z_1}$ | 1.9776 |
| $\mathbf{z_2}$ | 0 |
| $\mathbf{y_1}$ | 3.16 |
| $\mathbf{y_2}$ | 3.755278 |
| Min. objective | 3.18339 |

The above results show that SAND formulation converges to the same optimum value, and this required 22 iterations, 197 function evaluations, and 0.29 seconds [15]. The plot of function value with iterations is as shown below:

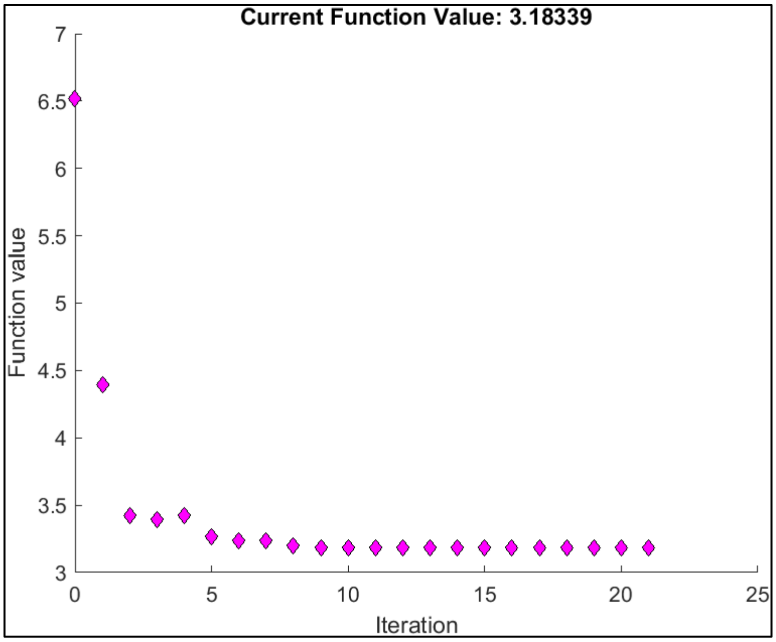



**Figure 2.3: Objective value vs no. of iterations for SAND**

The same problem can also be solved using ATC. The formulation used for ATC implementation of the sellar problem is as shown below:

System level optimization:

$$\min_{z_{11},z_{21},x_{11},y_{11},y_{21}} x_{11}^2 + z_{21} + y_{11} + e^{-y_{22}} + ALP(\phi_{12}) + ALP(\phi_{13})$$

$$\text{subject to: } \frac{y_{11}}{3.16} - 1 \geq 0$$
$$1 - \frac{y_{21}}{24} \geq 0$$
$$-10 \leq z_{11} \leq 10$$
$$0 \leq z_{21} \leq 10$$
$$0 \leq x_{11} \leq 10$$

Discipline 1:

$$\min_{y_{12},z_{12},x_{12},z_{22},y_{22}} ALP(\phi_{21})$$

$$\text{subject to:}$$
$$\frac{y_{1q}}{3.16} - 1 \geq 0$$
$$1 - \frac{y_{22}}{24} \geq 0$$
$$y_{12} = (z_{12}^2 + x_{12} + z_{22} - 0.2y_{22})$$

$$-10 \le z_{12} \le 10$$
$$0 \le z_{22} \le 10$$
$$0 \le x_{12} \le 10$$

Discipline 2:

$$\min_{y_{13}, z_{13}, x_{13}, z_{23}, y_{23}} \mathrm{ALP}(\phi_{31})$$
subject to:
$$y_{23} = \left(\sqrt{y_{13}} + z_{13} + z_{23}\right)$$
$$\frac{y_{13}}{3.16} - 1 \ge 0$$
$$1 - \frac{y_{23}}{24} \ge 0$$
$$-10 \le z_{13} \le 10$$
$$0 \le z_{23} \le 10$$
$$0 \le x_{13} \le 10$$

**( 11 )**

In the above formulation,

$$\mathrm{ALP}(\phi_x) = \mathbf{v}_x^{\mathrm{T}} . \phi_x + \|\mathbf{w}_x \circ \phi_x\|_2^2$$

$$\phi_{xy} = (\boldsymbol{x} - \boldsymbol{y})$$

The hyperparameters used in this formulation are $\gamma = 0.4$, and a dynamically updated value for β. This is an enhancement to the basic update strategy that was made to improve performance. As the sum of consistency and residual constraints decreases the β is also decreased. The exact update formula that determines the β at $k$th iteration is given by:

$$\beta_k = 1 + \mathrm{residual}_{k-1}$$

**( 12 )**

With this formulation the resulting optimal values are given in the table below:

**Table 2.3: Optimal values of the Sellar problem using ATC formulation**

| Design variable | Value |
|---|---|
| $\mathbf{x_1}$ | 0 |
| $\mathbf{z_1}$ | 1.9781 |
| $\mathbf{z_2}$ | 0 |
| $\mathbf{y_1}$ | 3.16 |
| $\mathbf{y_2}$ | 3.7562 |
| Min. objective | 3.1834 |

The ATC algorithm required 26 sec to converge after 90 iterations.

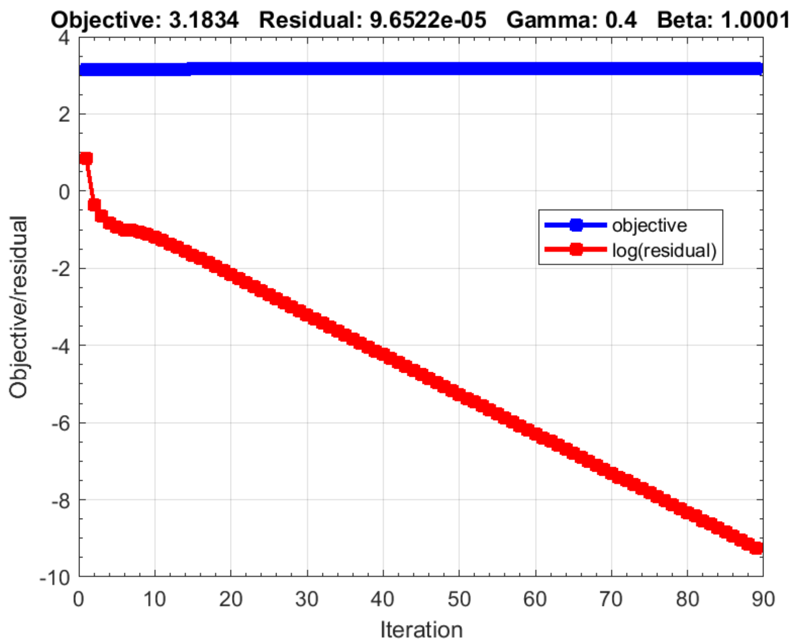



**Figure 2.4: Objective function & Residual vs iterations for ATC implementation of sellar problem**

In conclusion, both the ATC and SAND formulations converge to approximately the same result. However, ATC is noticeably slower and requires many iterations to converge to the optimal value. Another challenge faced when using ATC is that it is highly sensitive to the hyper parameter values $\beta$ and $\gamma$. To demonstrate the sensitivity of ATC with respect to the hyperparameters, different runs of this algorithm are executed with different combinations of $\beta$ and $\gamma$ values. The resulting optimal objective function value is tabulated below:

**Table 2.4: Sensitivity of ATC optimum to hyperparameters**

| Case | Initial $\boldsymbol{\beta}$ | $\boldsymbol{\gamma}$ | Min. Objective | No. of iterations |
|---|---|---|---|---|
| 1 | 1.2 | 0.4 | 3.184 | 87 |
| 2 | 1.5 | 0.4 | 3.184 | 54 |
| 3 | 3 | 0.4 | 3.184 | 60 |
| 4 | 1.2 | 0.6 | Does not converge | Does not converge |
| 5 | 3 | 0.6 | 3.43 | 69 |
| 6 | 3 | 0.3 | 3.184 | 54 |

Although the $\beta$ value that is provided to the algorithm is just the initial $\beta$, just the first iteration in which the $\beta$ is used itself has a significant impact on the computational expense. Likewise, some values of $\gamma$ make the algorithm completely unstable and unable to converge within the 200-iteration limit.

The reason for this is very similar to the sensitivity of step size in gradient descent solvers. If the step size is too high (analogously, if $\beta$ and $\gamma$ are too high), the solver can overshoot the optimal value and jump back and forth away from the optimal value as shown in image below

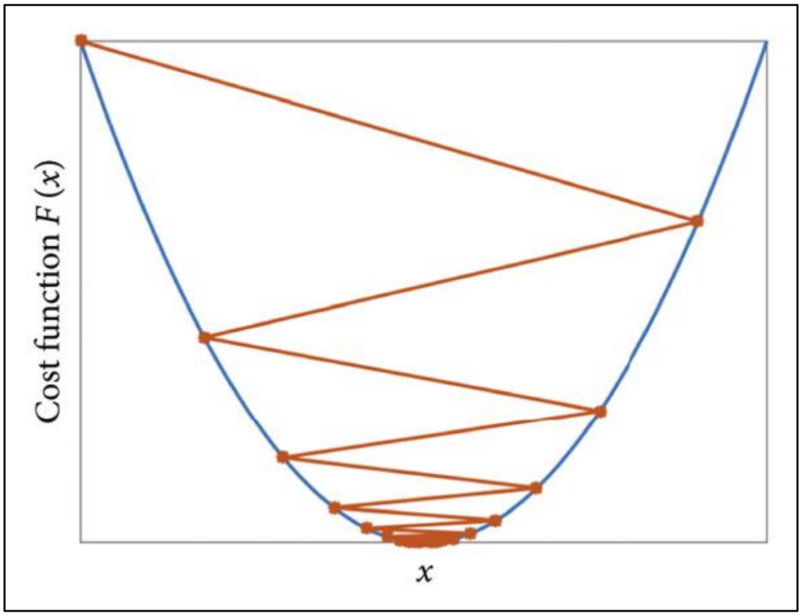



**Figure 2.5: Depiction of overshoot in gradient descent with high step size [17]**

Hence, the ATC penalty update algorithm must be tuned so that it can converge to the best result with a low computational expense. With this comparison between SAND and ATC now complete, we now proceed to formulate the eVTOL design problem, which can be optimized using the same ATC framework used for the Sellar problem. The sellar problem here serves as an evaluation criterion for testing the functionality and accuracy of ATC framework developed as a part of this study

# CHAPTER 3: EVTOL DESIGN OPTIMIZATION

The problem of designing an eVTOL was chosen to evaluate the feasibility of using ATC on complex engineering system optimization because of the richness of interactions that exists among its different subsystems. eVTOLs have also been gathering a lot of attention as a potential means of urban mobility in the future. eVTOLs aim to solve many existing drawbacks with the current urban mobility infrastructure by making commutes more environmentally friendly, affordable [19], and by relieving traffic congestion [17]. There are studies that show that even long range eVTOLs [20] are feasible, effectively making them comparable to helicopters for intercity air travel.

eVTOLs also come with their own set of challenges. For adoption of eVTOLs for urban transport, we also need to make sure that the noise levels generated by the aircraft are well within reasonable limits. This is particularly important, because unlike commercial long-range aircraft, eVTOLs are designed to fly at low altitudes.

The concept of eVTOLs originated with the need to develop an aircraft that is efficient in both lift and cruise phases of the flight, unlike traditional aircraft which are designed to be more efficient in the cruise phase. The reasoning behind this is that for short range operations, the aircraft will need to spend a significant portion of time hovering during take-off and landing and consequently, a larger portion of energy will be spent during these phases. Therefore, it is crucial that both these phases are efficient. As a result, many different configurations of eVTOLs have been developed. A few examples include tilt wing (vector thrust), multi-coptor, and tilt rotor. The figure below shows the different configurations of eVTOLs.

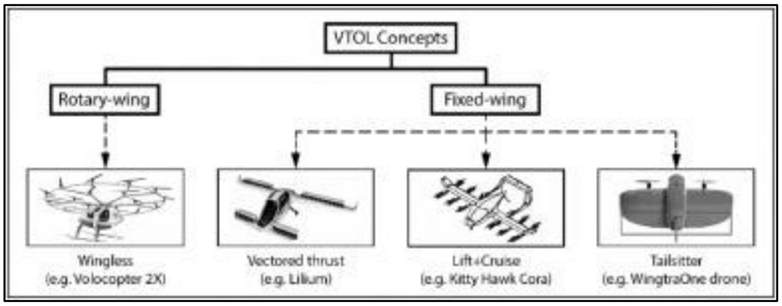



**Figure 3.1: Different types of eVTOL configurations [28]**

In this study we focus on tilt wing design with four rotors on each side. The following section covers the different models used in this study to analyze different components of the eVTOL system.

### 3.1 Motor model

There are four identical motors that power the rotors (propellers) on each side. All the motors in this model are sized using the maximum torque that the motor needs to generate [21]. The exact relationship used for sizing in this study is as follows:

$$m_{motor} = 0.03928\ T^{0.8587} \quad (13)$$

Here T refers to maximum motor torque. The aircraft configuration, including the number of propellers and rotors, used in this analysis is on Airbus Vahana eVTOL design [22]. The number of rotors is eight. The motor subsystem also computes the efficiency of the motor operating point based on an RPM-Torque-efficiency map. The problem formulation for eVTOL optimization is designed in such a way that the system level feeds the motor subsystem with angular speed (RPM) and torque and expects the subsystem to return an efficiency and mass value. The angular speed comes from a limit on the propeller tip speed:

$$\text{Tip speed}_{\text{propeller}} = \frac{2\pi}{60}\frac{r_{prop}\,.\,RPM}{V_0} < 0.65\ \text{Mach}$$

**( 14 )**

$V_0$ here is the speed of sound in air.

### 3.2 Propeller model

We discussed that the motor subsystem requires torque as an input. This input torque is computed in the propeller model. Using actuator disk theory, the power and the torque required can be calculated during hover if the thrust required is known. In the case of the hover phase, we know that the thrust needs to equal the weight of the aircraft. The equations below enlist the computation of hover power using actuator disk theory.

$$\sigma = \frac{n_{prop}\, c_{prop}\, r_{prop}}{\pi r_{prop}^2}$$

$$T_{hover} = \frac{W}{n_{prop}}$$

$$C_t = \frac{T_{hover}}{\rho_\infty \pi r_{prop}^2\, v_{tip}^2}$$

$$C_{pi} = \frac{C_t^{\frac{3}{2}}}{\sqrt{2}}$$

$$C_{p0} = \frac{\sigma C_{d0}}{8}$$

$$C_p = C_{p0} + kC_{pi}$$

$$\text{FOM} = \frac{C_{pi}}{C_p}$$

$$P_{hover,max} = n_{prop}\, T_{hover}\left(k\sqrt{\frac{T_{hover}}{2\pi\rho_\infty r_{prop}^2}}\right) + \sigma\frac{C_{d0}}{8}\frac{v_{tip}^2}{\left(\frac{T_{hover}}{\rho_\infty \pi r_{prop}^2}\right)}$$

$$Q_{hover,max} = \frac{P_{hover,max}}{\omega}$$

**( 15 )**

The solidity ratio for the propeller, σ, is chosen as 0.1, the blade airfoil drag coefficient, $C_{d0}$, is chosen as 0.012. $C_{p0}$ is the power required to overcome the profile drag of the airfoil and $C_{pi}$ is the ideal power needed to produce the required torque. k is an empirical correction factor that is used to correlate the actuator disk theory with experimental data [23]. $\rho_{\infty}$ here is the density of free stream air.

### 3.3 Wing model

Analyzing the wing is primarily important because it has a significant impact on the overall lift and drag of the aircraft. This analysis assumes a NACA airfoil (NACA 0012), and all the related parameters are taken from its datasheet. The whole aircraft lift coefficient $C_L$ is assumed to be 1.1 (including the wing and the fuselage) and $V_{stall}$ is assumed to be 35 m/s. Wingspan is then computed assuming two propellers per wing (two wings on each side). The outboard propellers are affixed at the wing tips and the distance between propellers is six times the propeller radius. The clearance between propeller hub and the fuselage is 1.2 m. With these assumptions, the maximum wingspan can be computed as:

$$b_{span} = 6.\, r_{prop} + 1.2$$

**( 16 )**

Since the surface area of the wing is assumed to be a design variable, the chord length of the wing can be computed as:

$$c_{\mathrm{wing}} = S_{\mathrm{wing}}/b_{\mathrm{span}}$$

**( 17 )**

The area drag of the fuselage, $SC_{\mathrm{d\ fuse}}$, is 0.35 and the span efficiency, e, is 1.3. The overall profile drag can then be computed as:

$$C_{d0} = C_{\mathrm{d\ wing}} + SC_{\mathrm{d\ fuse}}/S_{\mathrm{wing}}$$

**( 18 )**

The quadratic drag polar is then used to estimate the overall drag coefficient of the aircraft:

$$D = \frac{1}{2}\rho_{\infty} V^2 \left(S_{\mathrm{wing}}\left(C_{d0} + \frac{C_L^2}{\pi\ AR\ e}\right)\right)$$

**( 19 )**

Using the drag from the equation above, the power required to overcome this drag during cruise can be computed as follows:

$$P_{\mathrm{cruise}} = D\ V$$

**( 20 )**

Along with the computation of cruise power, the wing subsystem also computes the overall lift generated by the aircraft during cruise. This is constrained so that during cruise, the lift generated is equal to the weight of the aircraft.

$$L = \frac{1}{2}\rho_{\infty} V_{\mathrm{stall}}^2\ C_L\ S_{\mathrm{wing}} = \mathrm{MTOW}$$

**( 21 )**

MTOW here is the Maximum Take-Off Weight of the aircraft.

### 3.4 Gearbox model

The purpose of the gearbox is to enable the motor to operate at more efficient operating points. The gearbox is assumed to have a fixed efficiency value, which helps determines the power that is needed to be generated during the mission. The gearbox ratio can be fixed such that the power that needs to be generated by the motor is at an angular speed where its efficiency is a maximum (constant power curves on the RPM-Torque-Efficiency map). The Vahana aircraft does not use a gearbox as it adds additional weight. However, the argument made in this study (later shown to be correct under the assumptions made), is that using a gearbox increases the efficiency of the aircraft enough to reduce the required battery capacity and mass to offset gearbox mass. This tradeoff is clearly observed in the system optimization results which are shown in Chapter 4. The sizing of the gearbox is performed via an empirical model [24] that uses the power transmitted, motor shaft speed (RPM), propeller angular speed (RPM) to estimate the overall mass of the gearbox. The empirical relationship used here is:

$$m_{gearbox} = 0.453592\, k\, (hp^{0.76}) \frac{rpm_{motor}^{0.13}}{rpm_{prop}^{0.89}} \tag{22}$$

where hp is the power transmitted in HP and k is an index; an appropriate value for k here is 94.

**3.5 System level model:**

There are two options for system level objective functions that are often considered in eVTOL system design – total mass of the aircraft and cost per flight. The existing Vahana study implements the cost per flight as the objective function. This cost is a comprehensive sum that includes cost of operation, acquisition, fuel cost, and other factors. While cost per flight is a more accurate cost function that encompasses many factors and reflects overall system utility well, in the studies presented here the total mass of the aircraft is used as the objective to reduce computational complexity. With some preliminary optimization results, it can also be observed that minimizing both the cost and mass will yield approximately the same solution, so it can be argued that for system optimization studies at this level of fidelity, mass is an appropriate proxy objective function.

Within the system level ATC optimization subproblem, several constraints are computed that are not specific to any individual subsystem. Below is the list of constraints included in system level optimizer:

**Table 3.1: System level constraints in ATC formulation**

| Constrained variable | Constrained by | units |
|---|---|---|
| Battery capacity | Energy consumed in reserve mission | kWh |
| Mass of aircraft | Maximum take-off weight | kg |

### 3.6 Cruise Mission and Reserve Mission

The primary requirement for the aircraft in this optimization formulation is to perform a cruise mission as shown in the figure below. The mission includes a 90 second ascent phase, a cruise phase for the time required to cover a given range, and a 90 second descent phase. The ascent phase encompasses take off and acceleration phases where the aircraft ascends and then accelerates to cruise velocity. The aircraft has a transition phase where the aircraft increases its altitude and accelerates. However, for our computation purposes, the power consumed during this entire phase is equal to hover power (which is the maximum power that can be generated by the propellers). Similarly, the landing phase encompasses descent and deceleration phases. The cruise phase continues for a range of 50-150 km and the payload considered for this mission is 300 kg.

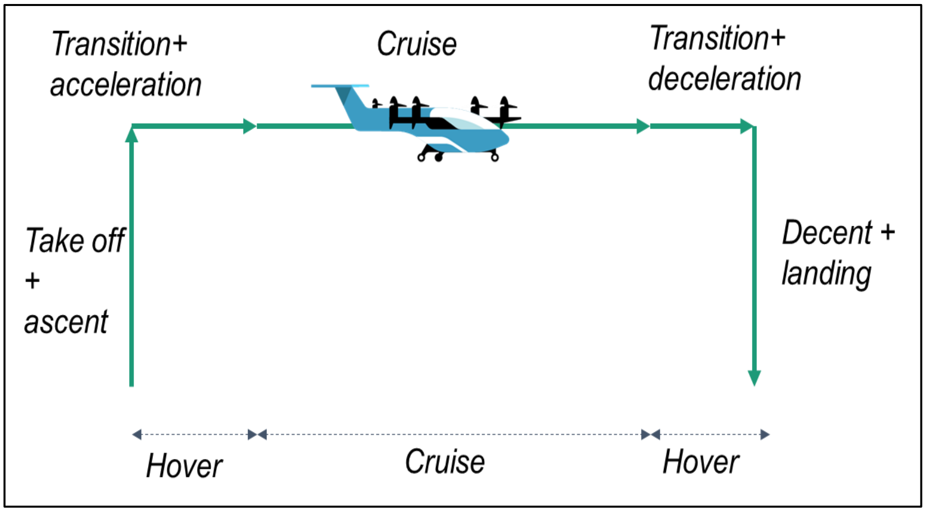



**Figure 3.2: eVTOL design optimization mission statement**

In addition to the cruise mission, a reserve mission is also modeled to include additional safety in aircraft operation. The reserve mission ensures that there is enough battery reserve charge such that the aircraft can loiter another 20 minutes at cruise velocity. Therefore, the battery capacity must be specified such that the aircraft can cruise for the given range and loiter for another 20 minutes.

**Table 3.2: Mission statement for eVTOL aircraft**

| Mission phase | Requirement |
|---|---|
| Take off and hover | 90 seconds |
| Cruise | [50,100,150] km |
| Reserve loiter | 20 min |
| Hover and landing | 90 seconds |

# CHAPTER 4: EVTOL ATC IMPLEMENTATION

The eVTOL design problem is optimized using a bi-level ATC architecture. The top-level optimization function includes the system level objective and constraints, and the bottom level includes the individual component analyses. The goal of the ATC formulation is to optimize eight design variables given in the below table.

**Table 4.1: List of design variables for eVTOL design optimization**

| Design variable | Units |
| --- | --- |
| Radius of propeller | m |
| Cruise velocity | m/s |
| Mass of battery | kg |
| Mass of motors | kg |
| Maximum takeoff weight | N |
| Wingspan | m |
| Angular speed in hover phase | RPM |
| Mass of gearbox | kg |

The component level optimization problem and the system level optimization problems both maintain a local copy of all associated design variables. These design variable copies ultimately converge to the same value (within a given tolerance limit) because of the ALP on consistency constraints. The objective for all the component level optimizers is just this ALP function, and their respective analyses compute the subproblem optimization constraints and the ATC responses needed for the penalty terms. A detailed schematic of the ATC formulation used in this study is provided in figure below.

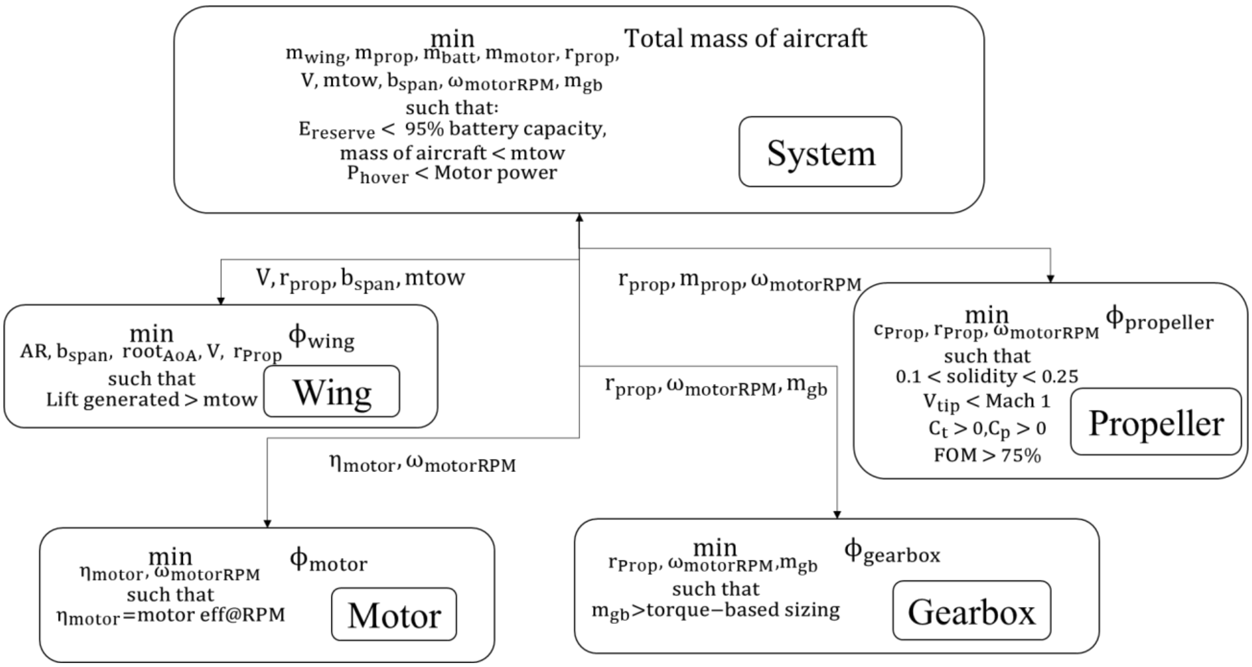



**Figure 4.1: Bi-level ATC formulation for eVTOL design**

To evaluate the performance and solutions of ATC optimization, the same problem is also formulated as a monolithic MDO problem. In this instance we use the Simultaneous Analysis and Design architecture (SAND) to compare the optimal solutions. In the SAND architecture, the penalty functions for the consistency constraints are not necessary as we only use one copy of design variables. All the other residual constraints are included as constraints of the single optimization problem with the objective being to minimize the total mass.

## 4.1 Design trade-offs based on the objective function

In Chapter 3 we discussed the topic of choosing an appropriate objective function. To understand the difference in results between choosing total mass or cost per flight as the objective function, different runs of SAND analysis were carried out. This was done in the initial stages of the project when the gearbox model was not yet included. Below are the figures showing the results of SAND using different objective functions.

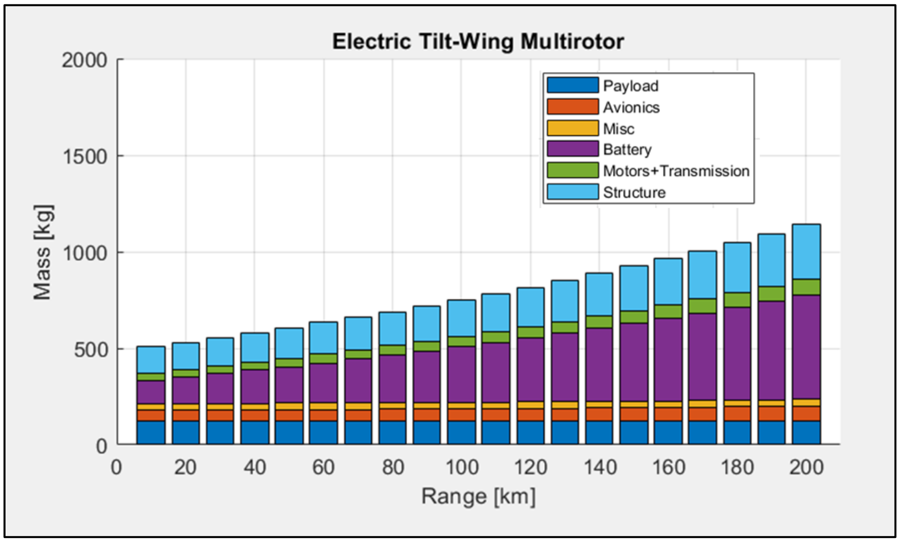



**Figure 4.2: SAND result with minimizing mass as objective function**

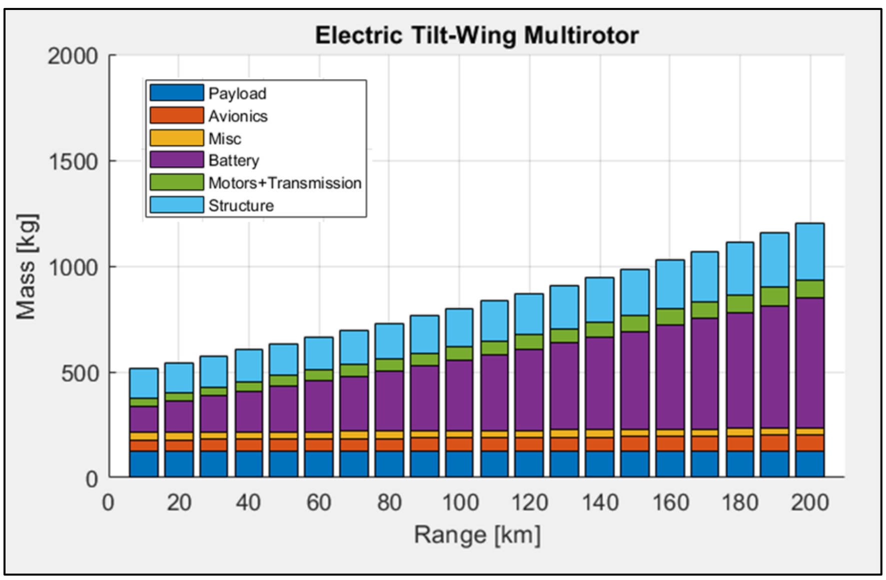



**Figure 4.3: SAND result with minimizing cost per flight as objective function**

From these results, we see that optimizing the total mass or cost per flight yields very similar designs with much of the difference in design choices occurring at higher range specifications. As we move towards a more cost-centric objective function, we see that:

- Area footprint is significantly reduced, primarily because propeller radius is decreased. This requires less space for maintenance, movement, and other cost drivers.
- A bigger motor and battery are used, thereby increasing the overall mass.
- Overall, the system might not seem to be operating at the most energy efficient point. Instead, it follows the route of reducing facility related costs to maintain a good cost efficiency.

However, the differences are relatively minimal, so using aircraft mass as a proxy objective function is a reasonable strategy to capture key trends at early design phases while requiring less computational effort.

### 4.2 Parameter sensitivity of ATC on the eVTOL problem

Earlier in the Sellar problem example we saw that the convergence and optimal solution of ATC are both highly sensitive to hyperparameter values for $\beta$ and $\gamma$. The impact of hyperparameters is also quite similar for eVTOL optimization using ATC. Higher values of $\beta$ and higher values of $\gamma$ generally make the algorithm overshoot the optimum and, in some cases, even cause algorithmic instability. The instability primarily arises out of the optimization problem becoming ill-defined due to extremely large penalty weights. Normalizing the weights and the system-level objective function does not seem to solve the instability. The effect of changing $\beta\ and\ \gamma$ can be observed in the figures below.

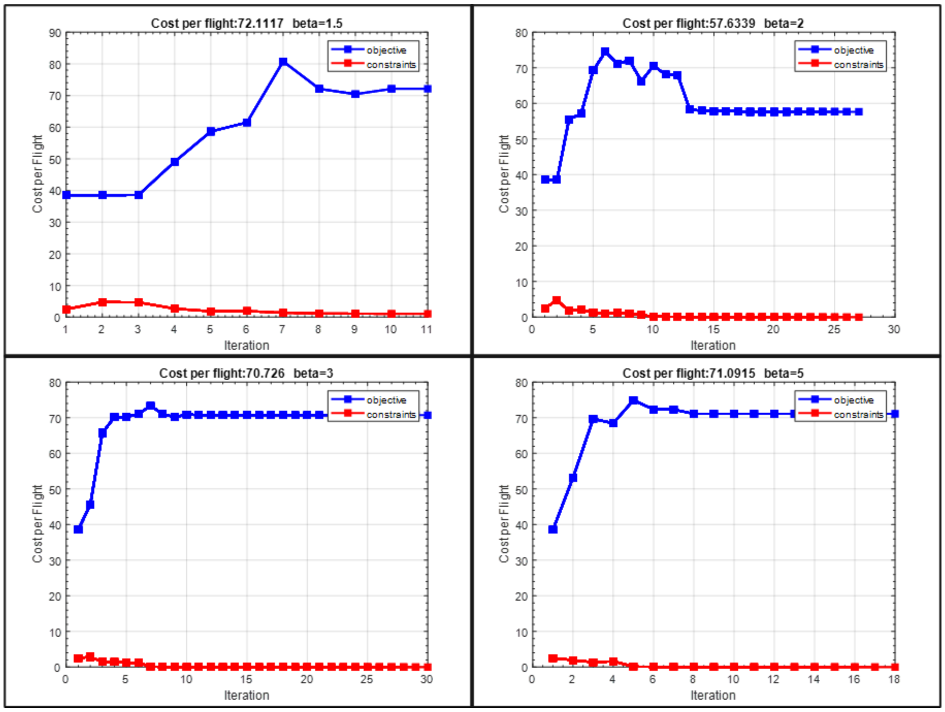



**Figure 4.4: Sensitivity of ATC to β**

We can see from the above figures that $\beta = 3$ and $\beta = 5$ solutions overshoot the optimum, which is found successfully when $\beta = 2$.

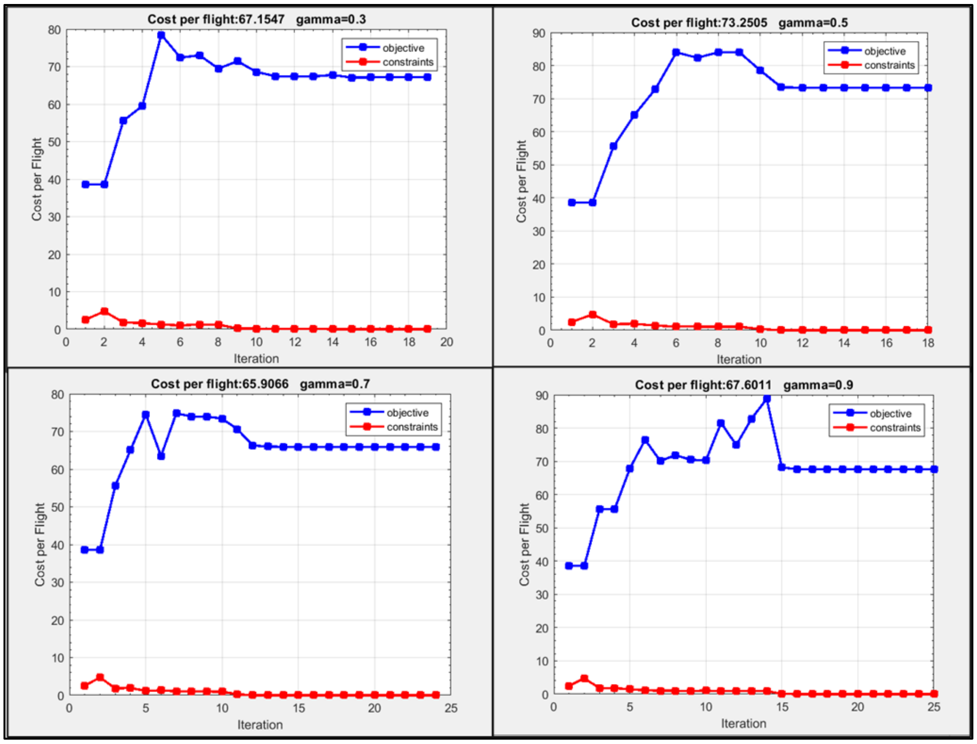



**Figure 4.5: Sensitivity of ATC to γ**

The sensitivity of gamma is still not well understood from the above figures. All the gamma values produce suboptimal results here.

## 4.3 Impact of including gearbox

The below results show the optimal cost for before and after including gearbox (payload: 100 kg, range: 50 km).

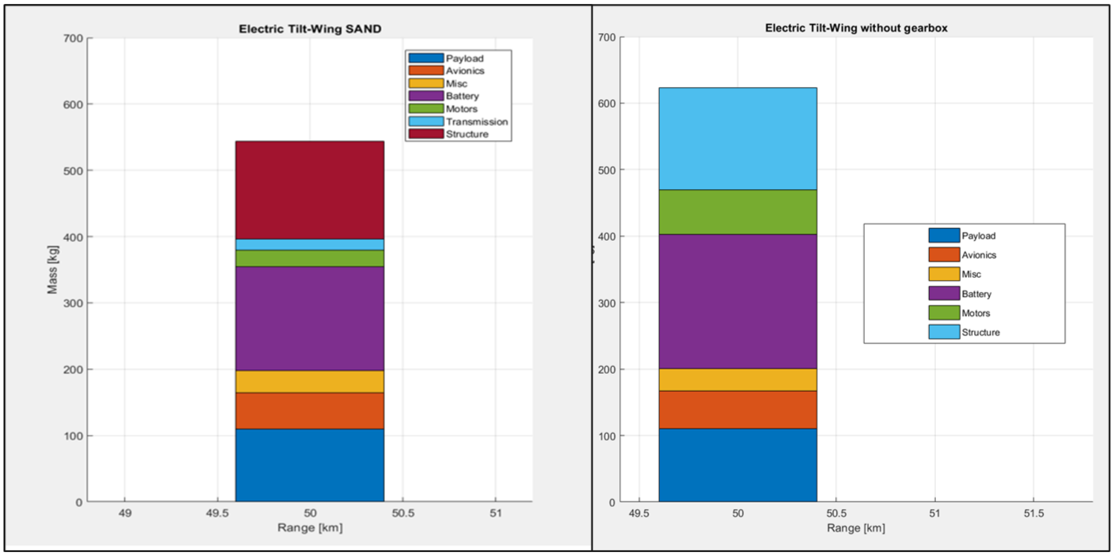



**Figure 4.6: SAND optimization results with (left) and without gearbox (right)**

The above results show that although the gearbox itself adds another 15 kg to the overall mass of the aircraft, it significantly reduces the mass of the required motor (from 60 kg to 20 kg) and reduces the required battery weight by about 50 kg. As a result, the overall aircraft is about 90kg lighter, further enhancing operational efficiency. Both the results shown in the above graphs have been optimized using SAND. This is the justification for including a gearbox in the ATC formulation.

### 4.4 Comparing ATC and SAND results

The main purpose of comparing ATC and SAND results is to understand the proximity of solutions using both the algorithms. For this study, the resulting solution of ATC is compared with SAND implementation over three different ranges. The figure below shows the results of this comparison.

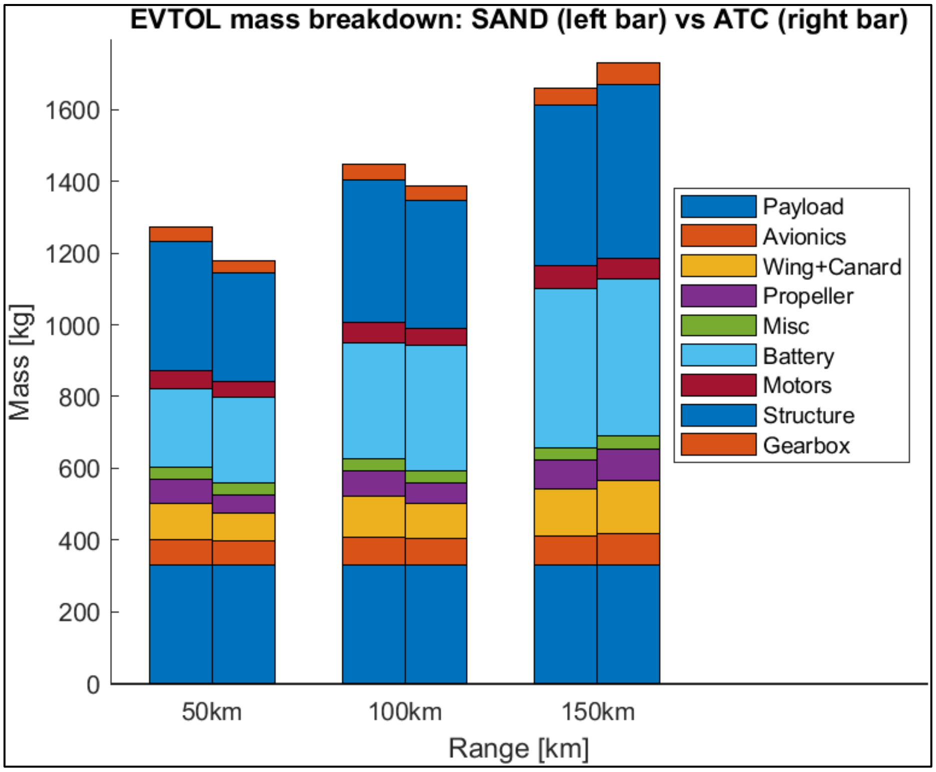



**Figure 4.7: SAND vs ATC optimal results comparison**

The details of the entries included in the legend are articulated in the table below.

**Table 4.2: description of legend entries in Figure 4.7**

| Legend Entry | Components included |
|---|---|
| Avionics | Servos, wire, tilt-mechanism |
| Wing | Wing and Canard |
| Propeller | Propeller and hub |
| Misc. | Seats, BRS |
| Structure | Structure and Fuselage |

As we see in the above figure, the optimal solutions of ATC and SAND are within 2-5%. These results can be reproduced with multiple (random) starting points for both the SAND and ATC implementations and hence, it can be concluded that ATC algorithm is capable of converging to the approximately the same solution as SAND. Something to be noted is that in previous literature, it has been proven that ATC can converge to the global optimum in the case of a convex problem [25]. Although we cannot guarantee anything about convergence to the global optimum in a non-convex problem, such as eVTOL design optimization (and most other practical engineering design optimization problems), we can observe that both the solutions converge to the same local optimum when the same starting point is used.  However, for some starting points, both the ATC and SAND algorithms struggle to find an optimum which is close to the local optimum found using other starting points. The figure below shows the optima found using different starting points.

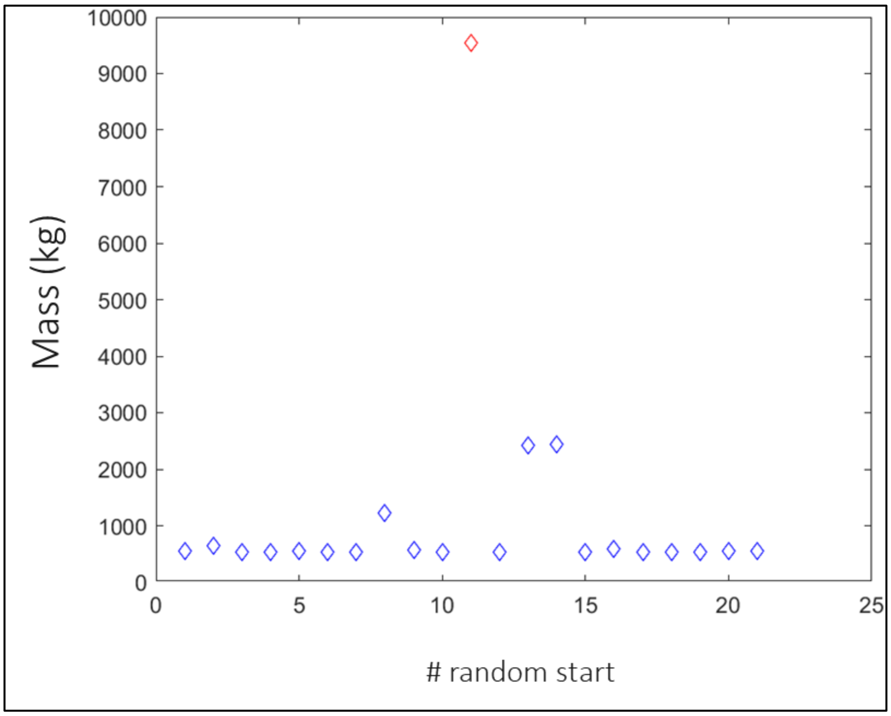


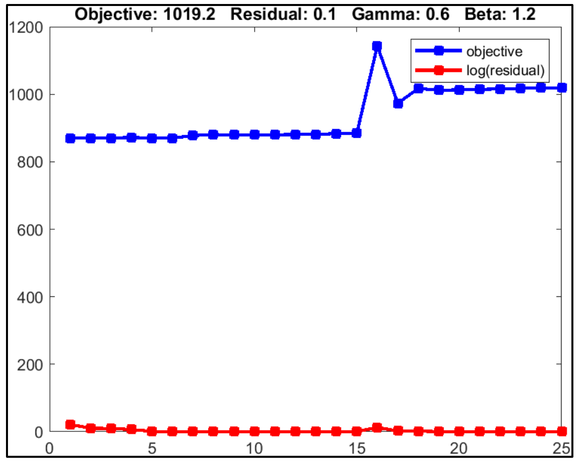

**Figure 4.8: Optimal objective value vs multiple restart points (SAND)**

The red data point in the above figure corresponds to an optimization run that did not converge to a local minimum, whereas the blue data points all correspond to optimization runs that did converge to local optima. Similar results can be observed with ATC, but in each of the tests the solution does not converge (satisfy consistency constraints to specified tolerances) after 200 iterations.

## 4.5 Comparing performance of ATC and SAND architectures

Performance comparisons are presented here to help understand the difference in computational effort between ATC and SAND architectures. One should keep in mind, however, that the disadvantages of ATC that are articulated here are only in terms of the computational cost. The overall effort required for conducting an engineering design optimization study requires for more than computation alone. As discussed in the next chapter, ATC may provide benefits for optimization studies being conducted by teams of engineers.

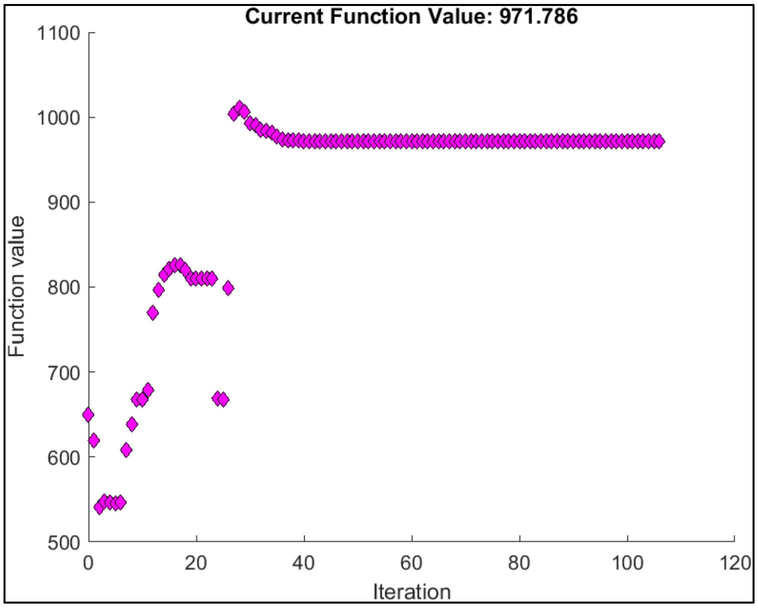
ATC requires an average of 320 seconds to complete for the 50 km case, and this required 25 iterations.




| Function Name | Calls | Total Time | Self Time* | Total Time Plot (dark band = self time) |
|---|---|---|---|---|
| motor_eta | 18312 | 218.141 s | 165.064 s | |
| griddata>useScatteredInterp | 18312 | 48.017 s | 48.017 s | |
| materials | 48156 | 37.226 s | 37.226 s | |
| wingMass | 24078 | 36.955 s | 12.516 s | |
| propMass | 12039 | 22.911 s | 8.099 s | |
| interp1 | 48156 | 6.261 s | 5.231 s | |
| pinv | 90817 | 3.688 s | 3.688 s | |
| fuselageMass | 12039 | 15.258 s | 3.197 s | |
| finDiffEvalAndChkErr | 59620 | 282.427 s | 2.936 s | |
| fun | 66391 | 7.120 s | 2.742 s | |

**Figure 4.9: Time profiling for ATC implementation for eVTOL design**

On the other hand, SAND takes about 56 sec and 106 iterations to converge for the same range and payload




| Function Name | Calls | Total Time | Self Time* | Total Time Plot (dark band = self time) |
|---|---|---|---|---|
| motor_eta | 2605 | 32.034 s | 24.312 s | |
| materials | 10424 | 8.547 s | 8.547 s | |
| griddata>useScatteredInterp | 2605 | 6.946 s | 6.946 s | |
| wingMass | 5212 | 8.845 s | 3.163 s | |
| propMass | 2606 | 5.407 s | 2.011 s | |
| interp1 | 10424 | 1.431 s | 1.196 s | |
| callAllOptimPlotFcns | 109 | 3.160 s | 1.058 s | |
| fuselageMass | 2606 | 3.562 s | 0.823 s | |

**Figure 4.10: Time profiling for SAND implementation for eVTOL design**

# CHAPTER 5: ATC HUMAN EFFORT ANALYSIS

In this chapter, the need for accounting for human effort in engineering design optimization studies is explained. As discussed in the previous chapter, the ATC (and distributed MDO methods, more broadly) often requires significantly more computational effort to reach the same solution accuracy as monolithic MDO methods. For the Sellar problem, ATC required ~50 times more time to converge, and for the eVTOL problem ATC required ~6 times more time to converge. Although this metric may not be an accurate way of demonstrating quantitatively the computational expense of ATC, it certainly shows that ATC has significantly higher computational expense compared to one monolithic MDO implementation. However, computational expense is only a part of the story when it comes to engineering practice. Often other phases of a design project such as interfacing with other engineering teams, revising the models used, etc., take up more effort. Quantifying all the human effort during the overall iterative problem formulation cycle is essential to understand the potential benefits of ATC.

## 5.1 Problem Formulation Cycle:

The problem formulation cycle describes the different stages of the iterative process required to successfully conduct design optimization studies. A schematic of this process is shown in the figure below.

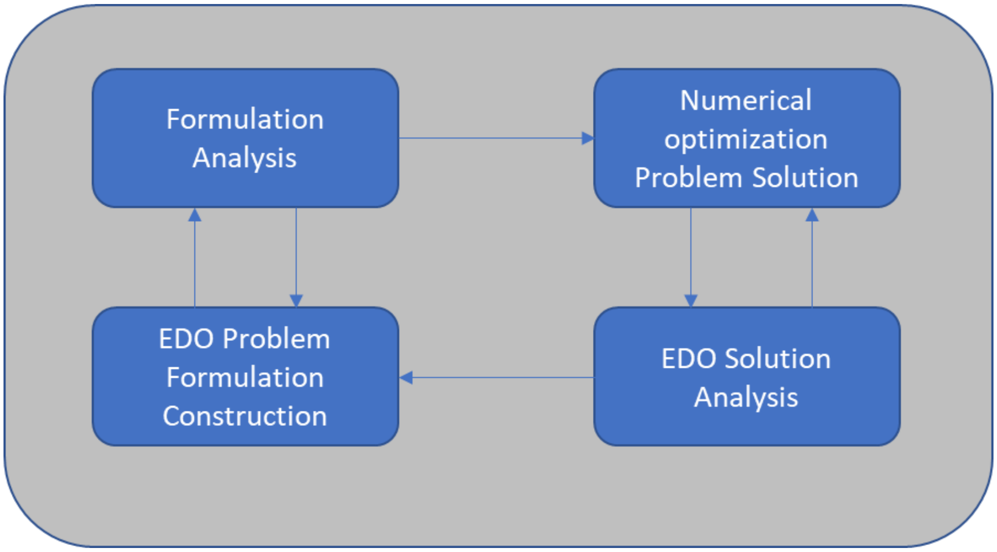



**Figure 5.1: Problem Formulation Cycle in Engineering Design Optimization**

In the problem formulation cycle, there are three different phases

### 5.1.1 Problem formulation

This is the first step in engineering design optimization. In this step the optimization problem is formulated by defining the design space, predictive modeling, and comparison metrics. There are a few types of analysis that we can perform right after problem formulation before even attempting optimization solution. For example, Monotonicity Analysis (MA) can be performed on the formulation to discover active constraints and perform problem reduction, as well as identifying potential formulation errors such as unbounded problems. If opportunities for improvement during formulation analysis are discovered, these can be implemented by revisiting problem formulation before proceeding to problem solution.

When attempting to execute MDO studies that involve multiple individuals or groups, ATC may offer benefits when considering the whole of the problem formulation cycle. Distributed MDO methods are modular and hence interchanging different models, or even adding/discarding new subsystems can be easier. In a monolithic MDO, changing the model may be disruptive and require a sequence of additional changes. If individual engineers or groups are focused on individual ATC subproblems, instead of needing to collaborate with the entire team on a shared problem formulation, individual ATC subproblems can be iteratively refined in parallel. This reduces the complexity of the communications required between engineers or engineering groups.

### 5.1.2 Problem Solution

In this stage, the optimization problem is already well formulated, and this is where the optimization problem is solved. During optimization, there is again a potential for iterations between different design choices and with each iteration, there is usually lot of communication required between different subsystems. That said, these interactions are computational as opposed to requiring human interaction. Although we have seen that ATC usually requires a greater number of iterations and function calls, yet the number of interactions is much more structured as compared to monolithic MDO methods and can take advantage of sparsity in system interactions. Especially with hierarchical ATC, the interactions are restricted to parent – child node communication and this limits the number of communication channels required. On the other hand, communication in monolithic methods is more open and includes communication channels between different subsystems. Thus, this stage can be computationally faster or slower for distributed MDO methods depending on factors such as the number of subsystems, effort required for each interaction, and other factors. Once these parameters are known we can model the complexity of this stage using different complexity metrics that exist in the literature [27].

A simple model for complexity of this stage is from Ref. [27]. The model complexity is given by

$$C = \sum \alpha_i + \left[\sum\sum \beta_{ij} A_{ij}\right] \gamma E(A) \tag{23}$$

where $\alpha_i$ is complexity of i$^{th}$ component, $\beta_{ij}$ is complexity of ij$^{th}$ interface, $E(A)$ is the sum of singular values of adjacency matrix(A), and $\gamma = \frac{1}{n}$, where n is number of individual components Using this model for ATC would yield:

$$C_{ATC} = n_{ATC}\left(\sum \alpha_i + n_{ALP} \sum \beta_{ij}\right) \tag{24}$$

where $n_{ATC}$ is the number of ATC cycles and $n_{ALP}$ encompasses the effort required forn communicating between two nodes of the ATC.

### 5.1.3 Solution analysis

This is the final step of design optimization in which the design results are analyzed. This step not only includes the analysis of final optimal solution but also includes analysis of trade studies and sensitivity analysis. With ATC, the sensitivity analysis can be easier to perform given that all the subsystems function in isolation and the effect of all the design variables (including local variables) can be analyzed.  The penalty weights of ATC can also be used to understand the sensitivity of residual and consistency constraints.

# CHAPTER 6: CONCLUSIONS AND FUTURE WORK

This study demonstrates the capability of ATC as an effective tool for design optimization of complex system of systems. Although it can be more computationally expensive than other MDO options, implementation of ATC in engineering design practice might decrease the overall combined effort (human plus computational) considering the entire problem formulation cycle as discussed in the previous chapter. This can only be concluded once a quantitative study of human effort is done to determine the exact extent of these benefits and understand whether these benefits outweigh the additional computational expense.

There are also few other advantages which may not have been covered in the problem formulation cycle section above. Distributed MDO also enables parallelization of workload in engineering design. In conventional analysis methods used in design optimization, there is a high level of interdependency. For example, in fixed point iteration, all the disciplinary analysis are executed in a sequential manner and then the loop is repeated. A rough schematic of this algorithm is shown in figure below.

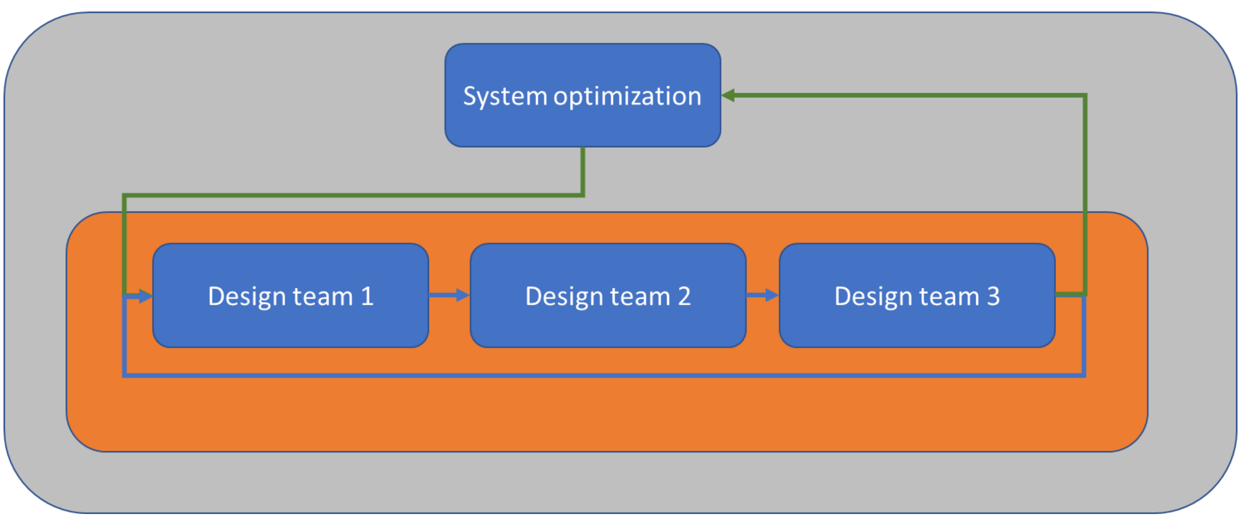



**Figure 6.1: Schematic for Fixed Point Iteration**

The blue arrows in the above figure indicate fixed point iteration and the green arrow indicate system level target and response propagation.

However, in hierarchical ATC, all the disciplinary analyses can be carried out in parallel as there is no interdependency between two subsystems of the same level. The interaction between two subsystems is via the parent node which is not executed until the next cycle. A rough schematic of ATC execution in engineering practice is as shown below

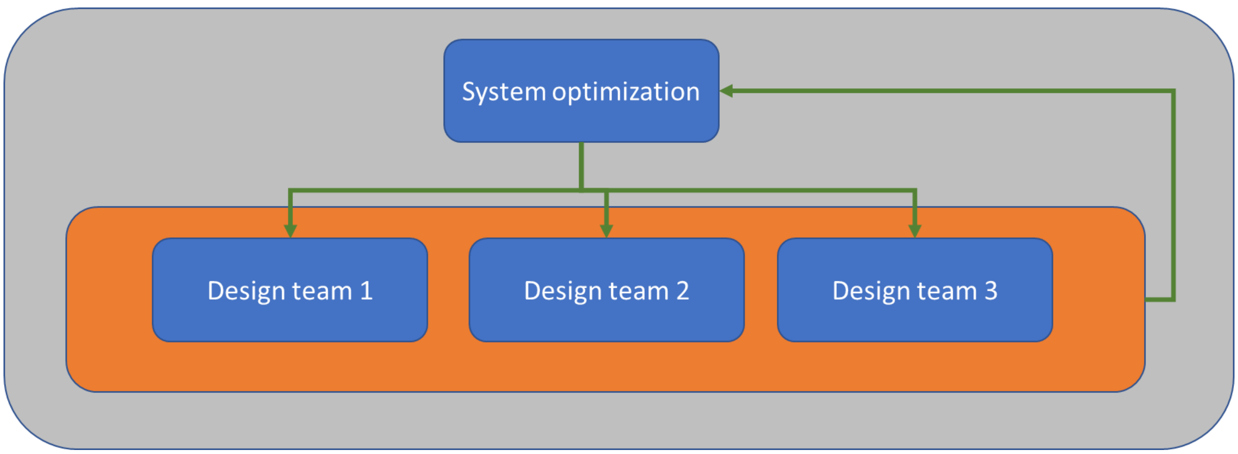



**Figure 6.2: Schematic for ATC**

ATC and all other distributed MDO methods also enable the design teams to choose which optimization algorithm is ideal for each subproblem. For example, some disciplines might have specialized algorithms available for efficient solution, such as optimal control. This is possible in a distributed MDO architecture because each discipline has its own optimizer. This is not possible in monolithic MDO methods because the entire problem is formulated as one single optimization problem.

## 6.1 Future work

This study lays out the necessity of human effort modeling of ATC implementation in engineering practice. A quantitative human effort model of the specific interactions, potential iterations and design changes is required to understand the exact criteria for which implementation of ATC can be beneficial. This may not make sense in a computational optimization setting as in this study but can be relevant for design optimization in engineering practice.

## 6.2 Conclusion

To summarize, we can conclude that ATC has the potential to be used in engineering practice for solving complex system of systems optimization problems. This is demonstrated by the proximity of optimal solutions of both ATC and SAND architectures with multiple random start points. It is intended that the work presented in this study and the arguments made for development of a human effort model for ATC implementations will pave the way for further research on the potential of ATC implementations in engineering design problems.

## REFERENCES

[1] Agte, Jeremy, et al. "MDO: assessment and direction for advancement—an opinion of one international group." Structural and Multidisciplinary Optimization 40.1: 17-33 (2010).

[2] L A Schmit. “Structural Design by Systematic Synthesis”. In 2nd Conference on Electronic Computation, pages 105–132, New York, NY, ASCE (1960).

[3] Swaminathan, Niraja, et al. "Flying Cars and eVTOLs-Technology Advancements, Powertrain Architectures and Design." IEEE Transactions on Transportation Electrification (2022).

[4] Pascioni, Kyle A., et al. "Acoustic Flight Test of the Joby Aviation Advanced Air Mobility Prototype Vehicle." 28th AIAA/CEAS Aeroacoustics Conference (2022).

[5] Hofacker, Cat, and Alyssa Tomlinson. "Building vertiport cities." AEROSPACE AMERICA 59.6: 22-32 (2021).

[6] Phero, Graham C., and Kessler Sterne. "The aerospace revolution: development, intellectual property, and value." (2022).

[7] Simpson, Timothy W., and Joaquim RRA Martins. "Multidisciplinary design optimization for complex engineered systems: report from a national science foundation workshop." Journal of Mechanical Design 133.10 (2011).

[8] Drew, C. “A Dream Interrupted at Boeing,” The New York Times, 5 September (2009), http://www.nytimes.com/2009/09/06/business/06boeing.html?_r¼1&hp (2009).

[9] Michelena, N., Kim, H., and Papalambros, P., “A System Partitioning and Optimization Approach to Target Cascading,” Proceedings of the 12th International Conference on Engineering Design, Munich, Germany (1999).

[10] Kim, H., Michelena, N., Papalambros, P., and Jiang, T., "Target Cascading in Optimal System Design," Journal of Mechanical Design, Vol. 125, September 2003, pp. 474–480 (2003).

[11] S Tosserams, L F P Etman, and J E Rooda. Augmented {Lagrangian} Coordination for Distributed Optimal Design in {MDO}. International Journal for Numerical Methods in Engineering, 73:1885–1910, 2008. doi: http://dx.doi.org/10.1002/nme.2158 (2008).

[12] Hyung Min Kim, Nestor F Michelena, Panos Y Papalambros, and Tao Jiang. Target Cascading in Optimal System Design. Journal of Mechanical Design, 125(3):474–480, September 2003. doi: http://dx.doi.org/10.1115/1.1582501 (2003).

[13] Tosserams, S., Etman, L. F. P., and Rooda, J. E., "Augmented Lagrangian Coordination for Distributed Optimal Design in MDO," Int. J. Numer. Methods Eng., 7313, pp. 1885–1910 (2008).

[14] Sellar, R. S., Batill, S. M., and Renaud, J. E., "Response Surface Based, Concurrent Subspace Optimization for Multidisciplinary System Design", 34th Aerospace Sciences Meeting and Exhibit, Aerospace Sciences Meetings, (1996).

[15] OpenMDAO 1.7.3 Alpha documentation. https://openmdao.readthedocs.io/en/1.7.3/usr-guide/tutorials/sellar.html (2021)

[16] Prajwal C, Sellar problem optimization using SAND and ATC https://github.com/chinthojuprajwal/Sellar_SAND (2022)

[17] Liu, Zhipeng, et al. "Gradient-sensitive optimization for convolutional neural networks." Computational Intelligence and Neuroscience 2021 (2021).

[18] Kleinbekman, Imke C., Mihaela A. Mitici, and Peng Wei. "eVTOL arrival sequencing and scheduling for on-demand urban air mobility." 2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC). IEEE (2018).

[19] Edwards, Thomas, and George Price. eVTOL passenger acceptance. No. ARC-E-DAA-TN76992 (2020)

[20] Alba-Maestre, Javier, et al. "Preliminary propulsion and power system design of a tandem-wing long-range eVTOL aircraft." Applied Sciences 11.23: 11083 (2021).

[21] Hendricks, Eric S., et al. "Multidisciplinary optimization of a turboelectric tiltwing urban air mobility aircraft." AIAA Aviation 2019 Forum. (2019)

[22] Graham Warwick. “Inside Airbus A3’s Vahana Electric VTOL.” In: Aviation Week Space Technology, pp. 1–4. issn: 00052175. url: https://github.com/VahanaOpenSource/vahanaTradeStudy. (2017)

[23] J. Gordon Leishman. “Principles of Helicopter Aerodynamics”. In: Cambridge University Pres (2006).

[24] Brown, Gerald V., et al. NASA Glenn Research Center Program in high power density motors for aero propulsion. National Aeronautics and Space Administration Cleveland (2005).

[25] Han, Jeongwoo & Papalambros, Panos. A Note on the Convergence of Analytical Target Cascading with Infinite Norms. Journal of Mechanical Design - J MECH DESIGN. 132. http://dx.doi.org/10.1115/1.4001001 (2010).

[26] Zhang, Daiyu, et al. "Performance evaluation of MDO architectures within a variable complexity problem." Mathematical Problems in Engineering 2017 (2017).

[27] Sinha, Kaushik, and Olivier L. de Weck. “Structural Complexity Quantification for Engineered Complex Systems and Implications on System Architecture and Design.” doi: http://dx.doi.org/10.1115/DETC2013-12013 (2013).

[28] Melo, Sofia Pinheiro, et al. "Life Cycle Engineering of future aircraft systems: the case of eVTOL vehicles." Procedia CIRP 90 (2020): 297-302 (2020).