Padova University

Master thesis

Study of Driving Factors in Preliminary Design of Autonomous Systems

Francesco Feltrin

supervised by
Prof. Alessandro Francesconi

September 2016

Abstract

The aim of the project is to individuate and understand the factors that affect the preliminary design for autonomous systems the most, with a particular focus on standard commercial satellites. This allows the solution of typical trade-offs (such as increased subsystem efficiency versus increased mass) using analytical tools instead of numerical optimization. Furthermore it can be used to quickly discard unappealing architecture (reducing development time) and to prove or validate engineering intuition, accelerating phase A.

Optimal system design goes one step further and, given a set of predetermined tasks that need to be performed, identifies a single *optimum design*. Although promising, very few successful examples are available in literature and they all deal with simple industrial applications, within a tightly controlled environment.

This constitutes a significant barrier for its application to autonomous systems which have to survive extreme, and extremely varied environmental conditions (i.e. launch segment and the space environment), perform considerably more tasks and satisfy stringent mass optimization. Furthermore, definition of an optimum point over a multi dimensional space (such as the space of all the subsystem requirements values) is non trivial, as the weight to attribute to each parameter has to be chosen on case by case basis.

Instead, we propose to map subsystem requirements (such as authority of the ADCS, battery capacity, on board storage capacity etc) as function of mission requirements. Once we identify the set of all the systems that satisfy mission requirements, we can study common traits or use some preference function (cost function) to choose an optimum design.

By keeping isolated the optimization/preference function from subsystem requirements definition, we maintain separated the constraints that are given from physics and mission definition (such as an orbit that does not allow for certain power systems) and those induced by the optimization. This is useful as the latter are typically based on more dynamic parameters (such as estimated development cost for a particular subsystem and performance level), which might be more subject to change during detailed subsystem design.

Given a required output, for example the torque needed for a given manoeuvre, and depending on system architecture, minimum subsystem requirement might be set unequivocally or not. If the ADCS has only 3 Reaction wheels, the control law is unique,

if there are 4, we have one redundant DOF, which causes ambiguity.
In the second case, in order to directly link subsystem requirements to mission requests, we need to define a **control allocation strategy**. If all actuators use the same resource (e.g. current), a standard strategy is to minimize said resource consumption. If more resources can be used, we need to decide how to distribute the *burden*; for example, if all the reaction wheels are almost saturated, we need to decide whether to saturate one (and eventually which one) or just increase the momentum of each one. The traditional approach is to assign a set of more or less arbitrary weights or costs.

There are cases in which this strategy is justified, and an equivalence between two non homogeneous resources can be objectively stated. Then, we can obtain subsystem requirements directly from the mission requirement and optimal control. Said equivalence does not often exist.

We propose to study, on an instant by instant base, the effect of the consumption of one specific resource versus all others, and then actuate the consumption strategy that promises best system performances. Two simplified models have been studied:

1. An hybrid vehicle has the option to either drain the battery, consume fuel or use a combination of both to meet instantaneous torque demand. Both battery charge and fuel quantity are finite. When the system is no longer able to provide the required torque, (either because it has run out of fuel, current or both) the system is said to be non responsive.

2. A singly redundant ADCS system based on reaction wheels is examined. To produce the time dependent requested torque, the system can allocate control among the wheels. Level of saturation are tracked and when a reaction wheel is fully saturated, the system is no longer able to use it.

In both simulations, we pursued the goal of *maximum responsive time*, the elapsed time between simulation start and the instant after which the system is no longer able to provide the required torque. Responsive time is a scalar value that can be used as an objective measure for control allocation effectiveness. Analytical results have been found and optimum (in the sense of *responsive time maximizing*) strategies have been identified under some simplifying assumptions.

In fact, the effectiveness of such strategy depends on the degree to which output request is known beforehand. First, we examined the case of perfect information (request is a function of time, but is completely known from the start). Subsequently, we limited information to statistical knowledge (only the average request is known in advance). It has been found that optimal control allocation behaviour imitates that of an economy in which resources prices are set using scarcity principle (the more scarce a resource is, the more valuable it becomes).

Having developed an optimum strategy for maximum exploitation of on board resources, we can assume its use (therefore eliminating any ambiguity), and determine the amount of on-board resources needed for a *statistically known* mission requirement.

This result can be used to determine minimum necessary battery capacity, given eclipse duration. In a similar manner, we can determine the minimum Reaction wheel angular momentum necessary to deal with a periodic disturbance, on board data storage given link availability period etc.

Other minimum subsystem requirements, such as authority or bandwidth can be determined in similar ways from mission requirements. If each task (thermal control, power generation, data handling) is performed by a non redundant subsystem, this relationship is direct and unequivocal. If there is redundancy however, an allocation strategy must be chosen.

We simulated a simple bus with Thermal control, Power system, on board data handling and generic payload, which sets instantaneous requirements. Using an allocation strategy to minimize authority we performed a Monte Carlo analysis to explore various mission scenarios, different degree of redundancy and subsystem efficiencies.
Given the high number of parameters, more than 500 thousand simulations were performed to increase confidence in the results. For a real mission however, many of these variable would be fixed by external conditions, allowing for a much faster, possibly analytical, solution. Strong correlation between mission requirements and reduction in minimum authority was found. Authority requirements were considerably lower when redundancy was accounted for, with average authority reduction between 10 and 15% (for each subsystem, compared with the single actuator case).

Finally, to better appreciate the advantages of the proposed method, we applied it to a specific mission. To further specify the problem, we chose to pursued mass minimization. This optimization choice allows us to solve many trade offs, by providing quantitative assessment on which subsystem authority has the greatest effect. To implement the approach, we developed parametric models for each subsystem from a database of cubesat components. Results from this optimization place the average decrease in mass between 5 an 10 %.

Therefore we can conclude that the **added degrees of freedom featured in a redundant architecture can be used to reduce overall system requirements**, under the assumption of appropriate control allocation algorithms. System flexibility gives the algorithms more way in which to accommodate external requests, allowing for a case by case optimization, based on up to date information about demand, instead of an optimization at the design level.

Introduction: Control allocation and Design

We can think of a subsystem as a set of interconnected actuators and a set of laws to coordinate them. For example, in the case of the thermal control subsystem, a resistor (the actuator) would depend on the power provided by another subsystem (PS), and would be governed by a law stating how much current the two subsystem need to exchange. We can represent both these ideas using a network diagram as shown below. Every subsystem is a node that can convert some input in output with a known production function, the connections show what kind of relationship are possible and the laws will quantify each exchange among the allowed interaction. Some subsystem provide the initial input (such as batteries). Finally, the system is able to produce what is of interest, the system outputs, which are usually the end goal of the subsystem cooperation efforts.

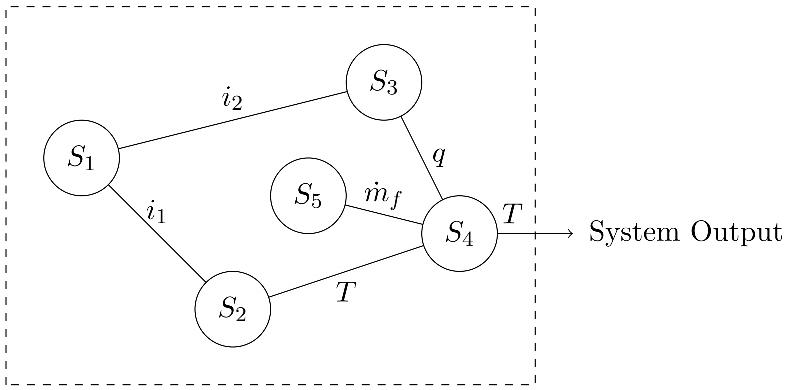


Figure 1: A system as a network of subsystems

Example 0.1. The network in figure 1 could be the representation of the architecture of a ground propulsion system. The battery S_1, provides current i_1 to the starter electric motor S_2 which converts it into torque T, starting the engine S_4. Once S_4 is self sustaining, it exchanges heat with the radiator S_3 (which might consume current as well) and it rotates the alternator, which will ultimately recharge the battery.

The above is **a set of subsystems** $(S_1, ..., S_5)$, **and a control allocation rule** which from the required torque T, is able to come up with the commands for every subsystem. From the control allocation standpoint, this approach shows how the same result might be achieved in many different way, which implicitly raises the question of which is better, hence **optimum allocation**.

To *design* the system and its parameters (such as the authority of each subsystem or their efficiency) we can use the network diagram to *compose* the production function of each component in that of the system. This however, is still dependent on the subsystem interaction laws.

The problem of control allocation/management can be solved independently from the problem of subsystem parameter design. In fact, we can operate under the hypothesis of a given design, and solve the **management**. In general however, there is no reason to assume the system to be bi injective; given a wanted output there usually is more than one way to produce it. This ambiguity is even more pressing in the case of Fault tolerant design, in which actuators are many times the number of degrees of freedom in order to assure operational continuity even after a failure event. To solve this, we will need to develop a concept of **optimal management**, or a way in which to express a preference.

Example 0.2. Redundancy and choice. Assume that the system in figure 1 is an hybrid car in which the internal combustion engine and electric motor are connected with a planetary gear. Upon the driver's request, the system must deliver a torque T. There are infinite ways to have the electric motors torque T_s and the internal combustion engine's T_{ic} add up to T. Even though they produce the same torque, it is clear that any combination of electric and combustion will have different secondary effects that we might want to consider. The **control allocation** algorithm will have to express a preference on the secondary condition, and then make the choice.

Once an optimal management algorithm is defined, we change the design (choice of subsystems) and find the new optimum management. By comparing the performances of the various systems under optimal management (hence *how well* they react to the same requests), we will be able to chose the system design that yields the locally optimum system.

Example 0.3. A racing car. It might be useful to apply this to the design of a racing car. The performance during the race will be the result of both the car design and the pilot skill (management).
With infinite budget, one can think to produce every possible car, have the same pilot race it, measure the time and hence chose the optimum vehicle design. While the design of the car could theoretically be guess work, the pilot must be consistent and possibly the best we have.
This naive optimization method might not be absurd if, instead of building the car, we use computer simulations.

Optimum Management

System management can be loosely defined as the set of instruction to give to every subsystem in order to obtain a preferred system behaviour. It should not come as a surprise that it heavily depends on the system design.

In this chapter, we will assume the system design to be given, from architecture to subsystem parameters, and examine a few types of basic system architecture types that can be used to solve the management of any system architecture/network.

We will begin with a formal definition of subsystem and the basic assumptions we will use in order to establish a common notation.

SISO Subsystem model

Definition 1.0.1. Subsystem
We define a subsystem as a component that allows management, that is a component in which the system management algorithm can directly control the input.

Example 1.4. A simple gear is not a subsystem because it can not be managed; It converts a torque T_1 into a torque T_2 according to its geometry

$$T_1 = \tau \cdot T_2$$

The gear ratio τ can not be changed by the management. Hence, we can not control/manage the power the gear produces by acting on the gear itself.

Example 1.5. If we put a clutch before the gear, we have a subsystem. We have to decide when to engage or disengage the clutch; we can perform a management by a crude PWM.

Example 1.6. A gearbox is also a subsystem; as we need to decide which τ to use among a few options.

The reasoning behind the previous definition is that, since we are interested in the management of the system, we don't need to care about components which, by definition, can not be managed. Furthermore, using this definition eventually each component will be part of a subsystem.

We will represent a subsystem as shown in figure 1.2, to emphasise that it consumes and produce some goods (g_0,g_1) and requires some form of management (x_i).

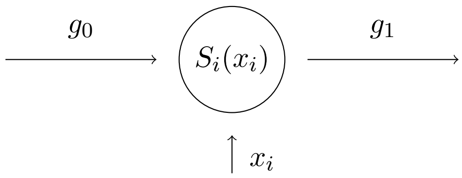


Figure 1.2: Siso Subsystem model

Since we can control this agent, there will be two functions

$$g_{0,S_i}(x_i) \qquad g_{1,S_i}(x_i)$$

that describe the relationship between the operational level x_i and the consumption/production. It is worth noticing that these functions are well defined.[1]. We can write a vector, called production plan

$$S_i(x_i) = \begin{pmatrix} g_{0,S_i}(x_i) \\ g_{1,S_i}(x_i) \end{pmatrix}$$

where, according with the usual notation, a negative value means consumption while a positive one represent production. We will call x_i the level of operation, which we assume always in $[0, 1]$ (or $[-1, 1]$ is the subsystem operations can be reversed), where 0 means that the system is turned off and 1 means that the system is working at maximum authority.

To simplify the notation, we will drop the subscript $_{S_i}$, and write $g_p(x_i)$ instead of $g_{p,S_i}(x_i)$ whenever this doesn't create ambiguity.

In general, we will assume that a SISO system either **simply produces** something without any input (like a battery or a fuel tank) or **transforms** some good in some other good. In the latter, we can see that in the space input-output (g_0, g_1), a system will work in the second or fourth quadrant, that is, consuming one resource to produce another.

[1]These are indeed functions, because we have designed the system to be controllable

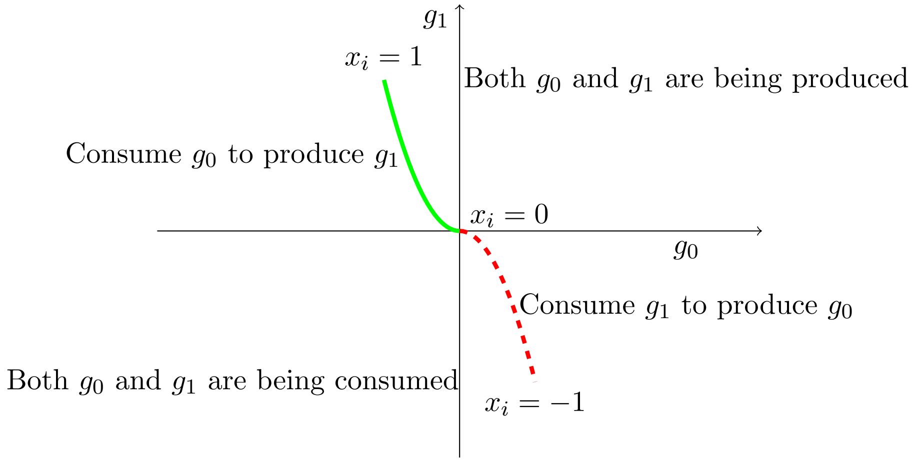


Figure 1.3: The production function of a SISO system in the input-output space.

Notes for the economic model;

These assumption will always be used unless otherwise specified; they are reasonable for engineers but are often explicitly stated in economic models of networks, and therefore might be interesting for economists

1. Possibility of inaction:
   We assume each subsystem can be turned off completely. That is $S_i(0) = \vec{0}$ for all subsystems.

2. Free disposal:
   We want to allow a subsystem to be running at sub optimal capacity, which means we allow a system to input more resources than those that it uses. [2]

3. Non decreasing production set[3];
   that is, by consuming more resources, the outputs can not decrease.

$$|g_0(\hat{x})| \geq |g_0(x)| \quad \Rightarrow \quad g_1(\hat{x}) \geq g_1(x)$$

   We can always assume that the level of operation x is monotonally linked to both production and consumption;

$$\hat{x} > x \quad \Rightarrow \quad |g_0(\hat{x})| \geq |g_0(x)| \text{ and } |g_1(\hat{x})| \geq |g_1(x)|$$

   since we have designed it this way.

[2]This is strictly not true for engineering systems however, we can assume that it is if we are sure that the economic solution we'll find will not use it (which is reasonable, as the maximum profit must be on the boundary of the production set) or if we assume that the exchange doesn't really happen. The subsystem pays for more input and disposes of it by letting it stays in the previous system.

[3]Since we allow for free disposal, the production function is extend to a production set

A more formal definition of management algorithm can now be introduced.

Definition 1.0.2. Management algorithm
Let S be an ensemble of n subsystems that operates with L goods; A management algorithm is an algorithm that, given the system requirements $\vec{R}$, produces the operational level vector X.

$$\begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{pmatrix} \in [0,1]^n \quad \text{such that} \quad \vec{S}(X) = \begin{pmatrix} g_{0,S_1}(x_1) + g_{0,S_2}(x_2) + \cdots + g_{0,S_n}(x_n) \\ g_{1,S_1}(x_1) + g_{1,S_2}(x_2) + \cdots + g_{1,S_n}(x_n) \\ \cdots \\ g_{L,S_1}(x_1) + g_{L,S_2}(x_2) + \cdots + g_{L,S_n}(x_n) \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ \cdots \\ r_L \end{pmatrix} = \vec{R} \in \mathbb{R}^L \tag{1.1}$$

Remark. Note that;

1. If a subsystem s takes no part in the production or consumption of a given good p, we have
$$g_p(x_s) \equiv 0 \qquad \forall x_s \in [0,1]$$
This means that usually the function $\vec{S}$ has a lot less terms than those shown above.

2. Usually, $n > L$; there are more subsystems that goods exchanged. This happens in redundant systems, where the number of actuators is greater than the number of outputs. This means that we have more unknowns than equations, and the solution, if it exists, might not be unique. **This is the problem of management**.

   If $n < L$ some goods would be coupled, that is, we could not produce (consume) one without producing (consuming) the other as well[4].
   If $n = L$, the system would be highly vulnerable to malfunctions; each subsystem failure (which we can model by taking out a subsystem production plan S_i) will result in a system failure (when the system is no longer able to meet the requirements $\vec{R}$).

For some system architectures, the problem of system management is *closed*; there is no ambiguity. For every system requirement $\vec{R}$ (system output) *with some reasonable choices*, we can determine the operational level X. We can write an injective function S.
We want to find out when this is the case, and what happens otherwise.

[4]This might still happen for some subsystems, but not for all. There are some instances in which this is tolerable, however, the more interesting case is the first one.

The reservoir subsystems

Since we have introduced a more formal definition of management algorithm, we want to give an idea of optimum management, which is the management algorithm that optimizes resource consumption.

The system depicted in figure 1.2 is an active component, or a subsystem that converts some amount of the good g_0 into some other amount of another good g_1. However, since we are mostly interested in autonomous systems[5] the resources are typically internal and therefore finite. Quite intuitively, at the beginning of each *productive chain* we have a reservoir subsystem that provides the initial good.
Examples of such subsystem might be a fuel tank, a battery, a reservoir of cool liquid for thermal control etc.

In order to facilitate the accounting of how much resources we are consuming with a given strategy, we can define an adjoint function similar to $\vec{S}(X)$,

$$\vec{Q}(X) : [0,1]^n \to \mathbb{R}^L \qquad \vec{Q}(X) = \begin{pmatrix} g_{0,S_j}(x_j) + g_{0,S_k}(x_k) + \dots \\ g_{1,S_p}(x_p) + g_{1,S_r}(x_r) + \dots \\ \dots \\ g_{L,S_m}(x_m) + g_{L,S_d}(x_d) + \dots \end{pmatrix} = Q_0$$

The only difference between $\vec{Q}$ and $\vec{S}$ is that the number of reservoir subsystem will be usually considerably lower than the number of active subsystem. We can expect typically only one reservoir subsystem for each good, (for example one set of batteries can be viewed as a single battery).

Remark. The agglomerate consumption function does not add anything; all the information we need are already present in the agglomerate production function $\vec{S}$. However, having a function that keeps track of all we are consuming can be useful to define some overall efficiency.

[5] Even if we are not, this still applies as the reservoir can be simply the subsystem that brings resources into the system.

Zero redundancy system; trivial management.

Imagine a series (or chain) of SISO subsystem, each one producing the good the following one needs.

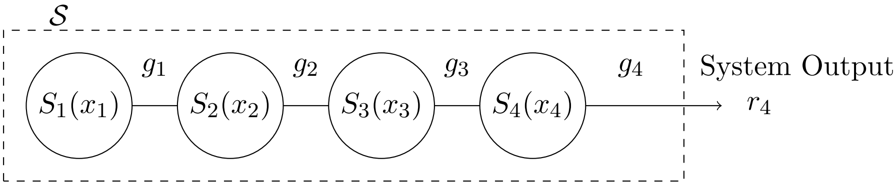


Figure 1.4: A Zero Redundancy System, simple chain or a series

To solve the management means to find X such that

$$X \in [0,1]^4 \quad \text{such that} \quad \vec{S}(X) = \begin{pmatrix} g_{1,S_1}(x_1) + g_{1,S_2}(x_2) \\ g_{2,S_2}(x_2) + g_{2,S_3}(x_3) \\ g_{3,S_3}(x_3) + g_{3,S_4}(x_4) \\ g_{4,S_4}(x_4) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ r_4 \end{pmatrix} = \vec{R} \in \mathbb{R}^L$$

The request is well posed if r_4 is in the range of possible output for the system; we assume

$$\vec{R} \in Im(\mathcal{S}) \quad Im(\mathcal{S}) \doteq \left\{ \vec{\rho} \in \mathbb{R}^L \text{ such that } \exists X \in [0,1]^n, \vec{S}(X) = \vec{\rho} \right\}$$

We now proceed to solve the management for a chain; There are two possibilities, depending weather all the g_i are invertible, or not.

1. All the g_i are invertible[6]
   There is no ambiguity in the network management. From r_4, inverting $g_4(x_4)$ we can determine x_4, hence x_3 and so on up to x_1. Formally the system is

$$\begin{cases} g_{4,S_4}(x_4) &= r_4 \\ g_{3,S_3}(x_3) &= -g_{3,S_4}(x_4) \\ g_{2,S_2}(x_2) &= -g_{2,S_3}(x_3) \\ g_{1,S_1}(x_1) &= -g_{1,S_2}(x_2) \end{cases} \Rightarrow \begin{cases} {g_{4,S_4}}^{-1}(r_4) &= x_4 \\ {g_{3,S_3}}^{-1}(-g_{3,S_4}(x_4)) &= x_3 \\ {g_{2,S_2}}^{-1}(-g_{2,S_3}(x_3)) &= x_2 \\ {g_{1,S_1}}^{-1}(-g_{1,S_2}(x_2)) &= x_1 \end{cases}$$

2. Not all the subsystem function are invertible.
   We can reduce the hypothesis of invertibility to the weaker hypothesis of *non decreasing* subsystem production function. The concept of efficiency allows us to swiftly deal with the issue of ambiguity. Assume that figure 1.5, represents

$$S_4(x_4) = \begin{pmatrix} g_3(x_4) \\ g_4(x_4) \end{pmatrix}$$

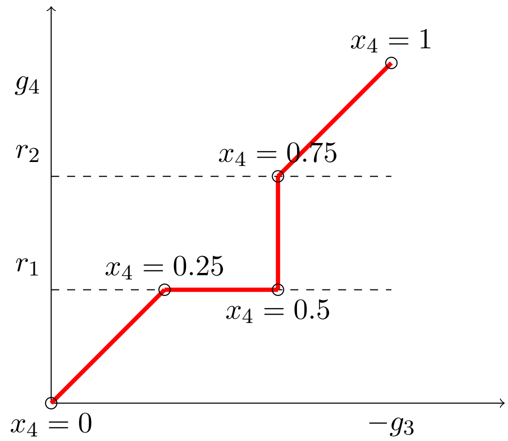


Figure 1.5: A patologically non invertible production function

When asked to produce the quantity r_4, any $x_4 \in [0.25, 0.5]$ would be a viable option. However, it is clear that $x_4 = 0.25$ is the best choice as it minimize consumption of g_3. The segment between $x_4 = 0.5$ and $x_4 = 0.75$ poses no problem. We can then define a new, invertible S_4 as

$$S_4'(x_4) = \begin{cases} S_4(x_4) & \text{if} \quad x_4 < 0.25 \\ S_4(0.25) & \text{if} \quad 0.25 \leq x_4 \leq 0.5 \\ S_4(x_4) & \text{if} \quad x_4 > 0.5 \end{cases}$$

S_4' is now invertible and we can proceed as in the previous case.

Note that:

1. We are assuming the system to be a productive chain, therefore we try to minimize consumption. If this were a *disposal* chain, our goal would be to consume as much as possible, hence we would pick the point that maximizes consumption.

2. We have obtained an invertible function, but $(g_4')^{-1}$ is not continuous. This can be overlooked as long as we are dealing with static requirements, but it might be a problem once we enter the realm of dynamics, unless we are dealing with zero order subsystems.

We formalize the above method with the concept of **engineering efficiency** or **strong efficiency**. We define the efficiency of a subsystem S_i as usual

$$\varepsilon_i(x_i) \doteq \frac{g_{p,S_i}(x_i)}{g_{p-1,S_i}(x_i)}$$

[6] We are implicitly relinquishing the option of free disposal

then, we can restrict an originally non injective function S_i to

$$S_i'(x_i) = \begin{cases} S_i(x_i) & \text{if} \quad g_p^{-1}(r_p) \quad \text{is unique} \\ S_i(x^\star) & \text{if} \quad g_p^{-1}(r_p) \text{ is a set} \quad \text{and} \quad x^\star \in g_p^{-1}(r_p) \text{ such that } \varepsilon(x^\star) \geq \varepsilon(x)\, \forall\, x \in g_p^{-1}(r_p) \end{cases} \tag{1.2}$$

Note; one can verify that $x^\star$ exists by proving the $g_p^{-1}(r_p)$ set to be closed and bounded (Extreme value theorem).

This maximizes system efficiency. Since all the production functions are increasing (non decreasing), using less resources in the a step will not use more resources in the previous. Then, this strategy can lead only to less or equal consumption.

Again, we want to point out that a zero redundancy system is highly vulnerable to malfunctions and losses in efficiency, as the system efficiency can be written as the product of the chain of efficiencies.

$$\varepsilon_S = \frac{r_4}{g_1(x_1)} = \frac{r_4}{g_3(x_4)} \cdot \frac{g_3(x_3)}{g_2(x_3)} \cdot \frac{g_2(x_2)}{g_1(x_1)} = \varepsilon_4 \cdot \varepsilon_3 \cdot \varepsilon_2$$

Note that:

1. Since the requirements are given (later we will see that these descend from the control theory), **maximizing efficiency means to minimize resource consumption**.

2. The efficiency of the system S_1 is not defined, as it is supposed to be a simple supplier.

We have solved the management of a generic chain, which now can be simplified as a single subsystem.

Non trivial management

The ambiguity in the system $\mathcal{S}$ will be on the combined production of S_1 and S_2. Hence, we will assume that each production function is already in its *invertible restriction*.

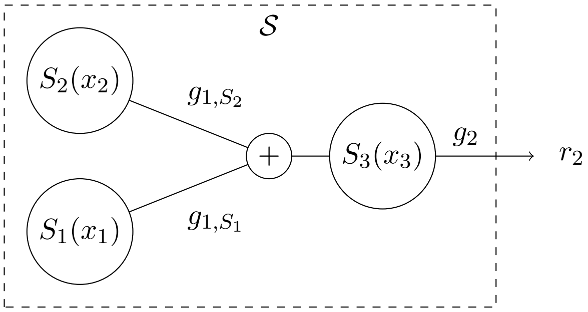


Figure 1.6: A Non-Trivial system with SISO subsystems

The management problem is to find

$$X \in [0,1]^3 \quad \text{such that} \quad \vec{S}(X) = \begin{pmatrix} g_{1,S_1}(x_1) + g_{1,S_2}(x_2) + g_{1,S_3}(x_3) \\ g_2(x_3) \end{pmatrix} = \begin{pmatrix} 0 \\ r_2 \end{pmatrix} = \vec{R} \in \mathbb{R}^2$$

The above system shows one degree of redundancy; the requirements of the subsystem S_3 can be met with any combination of $g_{1,S_2}+g_{1,S_1}$ such that $g_{2,S_3}(g_{1,S_2}+g_{1,s_1}) = r_2$.

Now that we have a mono dimensional choice, which is the best combination of $\mathbf{g_{1,S_2} + g_{1,S_1}}$? How do we chose?

We can differentiate between two cases; depending on whether or not S_1 and S_2 use the same resources. In the first case, the whole system can be still seen as a SISO system, while in the second, it has to be treated as a MIMO.

$\mathbf{S_1}$ and $\mathbf{S_2}$ homogeneous

In this case, both subsystems S_1 and S_2 transform the same input in the same output. The only presumable difference between the two system will be the respective efficiency. Notably, we can still define an **objective strong efficiency** for the first step, and the whole system as the ratio output input

$$\varepsilon_{\mathcal{S}} = \frac{g_{2,S_3}}{\text{Input}} = \frac{g_{2,S_3}}{f_i(X)}$$

Where $f_i(X) : \mathbb{R}^n \to \mathbb{R}$ is the function that counts the amount of the resource f_i consumed by the *reservoirs* (components such as a fuel tank, a battery, a pressure vessel).
By hypothesis, both S_1 and S_2 use the same resource, hence the denominator of the

efficiency is a scalar, and the definition is meaningful. We can then assume that the best solution will be that which maximises system efficiency $\varepsilon_{\mathcal{S}}$.

Example 1.7. Let S_1 and S_2 in figure 1.6 are two batteries that supply a current to the subsystem S_3. Even though nominally identical, they may exhibit different efficiencies due to their recent operational history; assume for example that S_1 is overheated, changing the chemical reaction effectiveness. We may chose to drain S_2 more than S_1, to obtain a better system efficiency.

Note that the efficiency $\varepsilon_{\mathcal{S}}$ is

1. **Meaningful**, because is a well defined fraction (we are not dividing by a multiple of zero) and we can always confront two option with their efficiency to find out which is best

2. **Objectively defined**, because **we can not distinguish** between the amount of resources that are used by S_1 or S_2, since they are of the same *kind/quality.*

To write a management function that maximizes efficiency, we can follow the process we used for a chain.

$\mathbf{S}_1$ and $\mathbf{S}_2$ not homogeneous

As mentioned before, we have to chose one element (X) for each R, so that we can invert the function $\vec{S}(R)$, and solve the management problem.

If the system deals with multiple goods, its consumption (input) has to be described with a vector in $\mathbb{R}^n$. Then we can no longer define an efficiency as the ratio of output input, because vector division is meaningless. We can still think to use a concept similar to *efficiency*; since the requirements (outputs) are given, we could just focus on minimizing consumption (input), without performing a division.

This leads to a somewhat similar problem; in $\mathbb{R}^n$ it is not obvious which among two element is the smaller.

Example 1.8. Let figure 1.6 be a heating system; S_1 is a resistor, while S_2 a hot liquid from which we extract heat. What should we use to heat the subsystem S_3? How can we compare two possible alternative? Are we using *less* resources if by using electric energy, the hot liquid, or a combination of the two?

There are two conventional options:

1. Since we can not compare two element in $\mathbb{R}^n$, we **arbitrarily define** a function $\mathcal{C} : \mathbb{R}^n \to \mathbb{R}$, so that we can define a total order on the image of the element of $\mathbb{R}^n$ (the usual order $\leq$).
   It is clear that this function can not be injective (due to the dimensions of the problem). Hence, we may find that the maximum value of the cost function $\mathcal{C}$ will be linked to many points of the domain, and the problem is still not completely solved.

Example 1.9. A consolidated approach for the function C is to assign a cost/price to each resource the system uses. The total cost of a strategy/choice can then be computed as

$$\mathcal{C}(X) = x_1 \cdot c_1 + x_2 \cdot c_2 + ... + x_n \cdot c_n = \vec{x} \cdot \vec{C}$$

The cost function can be seen as a simple dot product, once we have identified the cost vector $\vec{C}$ such that $\mathcal{C}(X) = X^T \cdot \vec{C}$.

$$\vec{C} = \begin{pmatrix} 1 \\ 5 \end{pmatrix} \quad \mathcal{C}(X) = 1 \cdot x_1 + 5 \cdot x_2$$

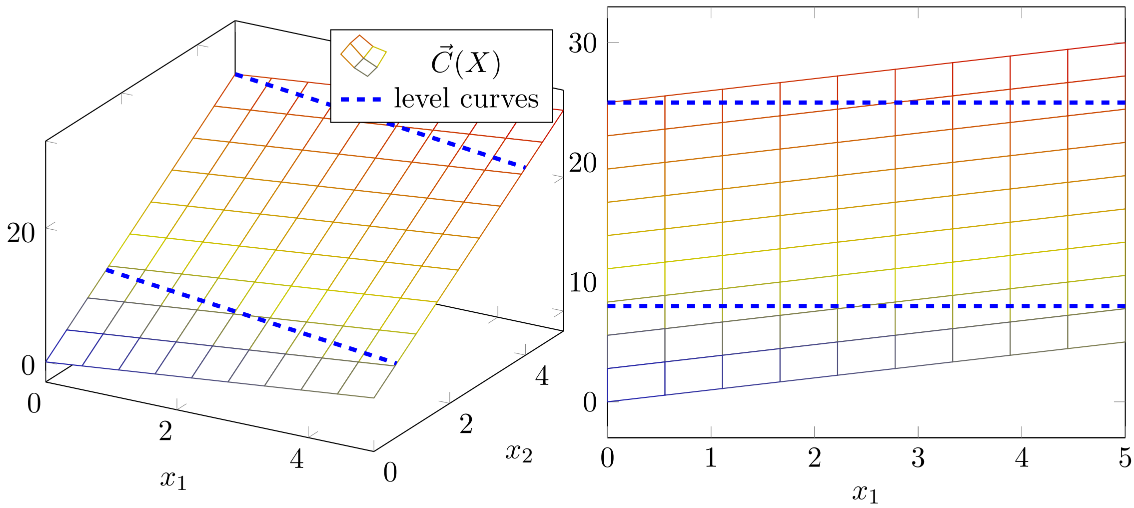


Figure 1.7: Cost function; level curves are points which are equally expensive, and therefore equally valid.

Example 1.10. Another idea is to use the norm, which can be seen as a particular cost vector dependent on the quantity itself ($\vec{C} = X$). One should not think of this as an objective choice just because it doesn't require an arbitrary cost vector. The relatives weight are implicit in the choice of the unit.
If we are comparing fuel mass and current, there is big difference between the norm of a vector written in kg or in grams!

Once a cost function is defined, finding the optimum value is an exercise that can be solved in various ways. Numerical solution can be used, Lagrange optimization and so on. However, the question remains on how to assign the cost vector in an objective way. This is crucial only for the the concept of *optimum* allocation or management, the system will operate with any cost vector but in a suboptimal way.

2. We accept a partial order.
   This means that we accept that there will be many solutions which we are not able to compare. Given two elements in $\mathbb{R}^n$ we may or may not be able to say which is bigger/better. This is quite similar to the above case except for the fact that we don't have to chose the price vector C. In this sense, this method is objective.
   In fact, one can prove that, if a maximum (minimum) exists for this partial order, it is also the maximum (minimum) *for any choice of price vector* $\vec{C}$.

Partial order definition

Definition 1.0.3 (Partial Order)**.** Let A be a set and $\sqsubseteq$ a relation on A. $\sqsubseteq$ is a partial order on A if it is

(a) Reflexive: $\forall a \in A$, $a \sqsubseteq a$

(b) Antisymmetric: $a, b \in A$, $a \sqsubseteq b$ and $a \sqsupseteq b \Leftrightarrow a = b$

(c) Transitive: $a, b, c \in A$, $a \sqsubseteq b$ and $b \sqsubseteq c \Rightarrow a \sqsubseteq c$,

Example 1.11. Partial order "divide exactly"
Let us define the *divide* order on the natural set $a, b \in \mathbb{N}$, as $a \sqsubseteq b$ if $\exists\, x \in \mathbb{N}$ such that $a \cdot x = b$.
We can check that this relation is reflexive, antisymmetric and transitive, in fact

(a) $\forall a \in \mathbb{N}$, $a \cdot 1 = a$

(b) If $\exists\, x \in \mathbb{N}$ such that $a \cdot x = b$ and $\exists\, y \in \mathbb{N}$ such that $a = y \cdot b$, then clearly $a = a \cdot x = b = b \cdot y \Rightarrow x = y = 1$ hence $a = b$.

(c) If $\exists\, x \in \mathbb{N}$ such that $a \cdot x = b$ and $\exists\, y \in \mathbb{N}$ such that $b \cdot y = c$, then $a \cdot x \cdot y = c$ hence $a \sqsubseteq c$.

Definition 1.0.4 (Total Order)**.** If, for every couple of elements in A, we can say either $a \sqsubseteq b$ or $b \sqsubseteq a$ (or both), the partial order is said to be a total order.

Example 1.12. The partial order in example 1.11 is not a total order. To prove this, note that 2 does not divide 3 and 3 does not divide 2.

Remark. Recognize that, on the vector space $\mathbb{R}^n$ the norm of a vector does not constitute a partial order as, in general, it is not antisymmetric (two vector can have the same norm and not be the same vector).

A partial order on $\mathbb{R}^n$ can be defined as

$$\vec{a}, \vec{b} \in \mathbb{R}^n \quad \vec{a} \ll \vec{b} \Leftrightarrow a_i \leq b_i \quad i = 1, 2, ..., n$$

Clearly, this is reflexive, antisymmetric and transitive. However this does not constitute a total order, hence there are elements of the domain that can not be compared with one another. This is known in economics as Pareto efficiency.

To understand the value of this definition, consider the following case in which we use it to compare the points A,B,C,D in $\mathbb{R}^2$

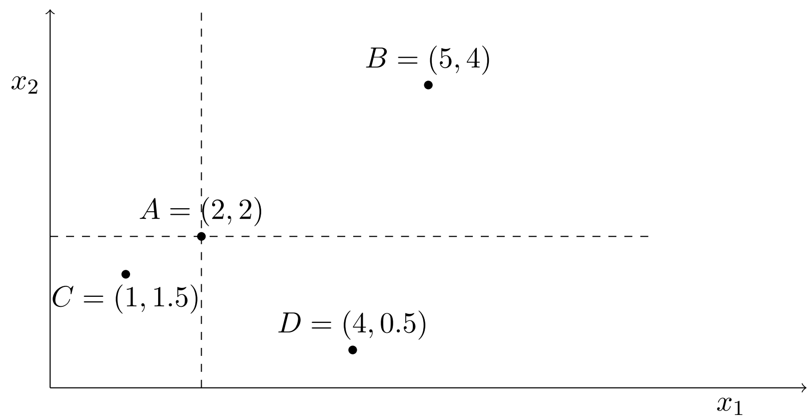


Figure 1.8: A partial order in $\mathbb{R}^2$

In figure 1.8 we can recognize that

$$C \ll A \quad A \ll B \quad D \ll B$$

But we can not say either $A \ll D$ nor $D \ll A$.

However, as mentioned before, this partial order has the propriety to be objective; notice that, if $A \ll B$ the cost of the operation A is less than that of the operation B regardless of the cost vector $\vec{C} \in \mathbb{R}^+$. In fact

$$\vec{C} \cdot A \leq \vec{C} \cdot B \qquad c_1a_1 + c_2a_2 + \cdots + c_ma_m \leq c_1b_1 + c_2b_2 + \cdots + c_mb_m$$

$$c_1(a_1 - b_1) + c_2(a_2 - b_2) + \ldots c_m(a_m - b_m) \leq 0$$

Which is true when we assume that $A \ll B$, $\vec{C} \in \mathbb{R}^+$. The partial order $\ll$ is a very strong condition, but it isn't always applicable.
Then, if A is a minimum (with regard to the $\ll$ partial order), it will be a minimum for every possible cost vector we could think of.

Finally, we can come up with more creative ways to establish a preference. We would like to have the best of both options, which is a cost function, which is also objective. This will be the topic of the next chapter.

A new approach to resource allocation strategies

In the previous chapter, we showed some methods to manage a resource consumption once a cost vector has been chosen. In this chapter we will explore the reasons we might use to chose such a vector.

Intuitively, the cost vector establishes an equivalence among our finite resources; It states, for example, that 1 g s^{-1} of a specific fuel might be equivalent to a current of 1 A . But how can we compare two non homogeneous quantities ?
In well defined circumstances, the laws of physics can establish (or at least suggest) an equivalence; we can compare different types of fuel on the basis of the chemical energy they released when burning. However, more than one parameter might be involved in the comparative evaluation; we might wish to compare densities, economic costs, reactivity (as in the capacity to oxidise the engine) etc.

To begin with, let us use the equivalences that are implicit in the system architecture itself. **An equivalence among different resources might be established using the production functions of subsystems with homogeneous output.**

Example 2.13. Reaction wheel and thrusters for Attitude Control
Assume we have a satellite which can control its orientation using both reaction wheels and thrusters. The total output torque T will be the sum of the torque given by both subsystems, $T_{tot} = T_{RW} + T_{Th}$. We can write each subsystem contribution as a function of its resource consumption; $T_{tot} = T_{RW}(i) + T_{Th}(\dot{m}_f)$. We can therefore establish and equivalence among the two inputs for each level of produced output

$$T' = T_{RW}(i') = T_{Th}(\dot{m}'_f) \qquad i' = T_{RW}^{-1}(T') \quad \dot{m}'_f = T_{Th}^{-1}(T')$$

It is not unreasonable to say that the current i' and fuel flow $\dot{m}'_f$ are **equivalent in their ability to produce the same torque** T'. The two quantities i' and $\dot{m}_f$ obviously have no reason to be the same; they will just cost the same

$$c_m \cdot \dot{m}'_f = c_i \cdot i' \quad \Rightarrow \quad \frac{c_m}{c_i} \doteq \frac{i'}{\dot{m}'_f}$$

We need to consider some issues

1. If the relative cost if determined using the same equation that states system output, we are not adding anything. This has the advantage of being objective, **but does nothing to solve the problem**. We are merely using the same equation twice. **The real cost choice will be something similar**, but not exactly as shown above

   Example 2.14. Assume that the production functions of example 2.13 are both linear; then

   $$T = \alpha i + \beta \dot{m}_f \qquad i' = \frac{T'}{\alpha} \quad \dot{m}_f' = \frac{T'}{\beta} \quad \rightarrow \quad \frac{c_m}{c_i} = \frac{\frac{T'}{\alpha}}{\frac{T'}{\beta}} = \frac{\beta}{\alpha}$$

   The equation $c_m \cdot \dot{m}_f = c_i \cdot i$ means that if we both increase a current byΔi and decrease fuel flow of $\Delta \dot{m}_f = \frac{c_i}{c_m} \Delta i$, we will end up paying the same price. The cost function is

   $$C = c_i \cdot i + c_m \cdot \dot{m}_f$$

   Then if we add the cost function $\vec{C} = (c_i, c_m)^t$ in order to close the problem of management we find that

   $$\begin{cases} T = \alpha i + \beta \dot{m}_f \\ C = c_i \cdot i + c_m \cdot \dot{m}_f \end{cases} \Rightarrow \quad \begin{cases} \frac{T}{\alpha} = i + \frac{\beta}{\alpha} \dot{m}_f \\ \frac{C}{c_i} = i + \frac{c_m}{c_i} \cdot \dot{m}_f \end{cases} \Rightarrow \quad C = \frac{T}{\alpha} \cdot c_i$$

   Hence, given a torque requirement, any choice of $i, \dot{m}_f$ will cost the same; We can not use this *exact* method to close the problem.

2. Once again, we require the production function to be invertible. We have shown how the hypothesis of maximum efficiency is enough to write an invertible restriction of a production function. However, consider that some values of T might be in the co domain of only one of the subsystem. In this case, we can think of the relative cost either as $c_m = \infty$ or meaningless.

3. If the production functions are linear, price is constant. Otherwise, it usually won't be.

Another cost definition strategy

As shown above, if we want to base our cost/equivalence on system architecture, we need to find a function which is independent from the system production function. A rather objective and sensible choice to define the costs would be to pursue **maximum system duration**. We want to find the cost function $\vec{C}$ that maximizes responsive time, which we define as the time in which the system is able to meet requirements.

Remark. The above goal is tailored for autonomous systems, but can be extended to almost any system. There are however, some notable exceptions:

1. Systems that have access to **unlimited external resources**.
   We can imagine an industrial system that is plugged in the electric grid for power, obtains water for cooling from a river and so on. In this instance, we don't need to find a cost vector, because real word economics does it for us. We have an objective cost function that we usually wish to minimize, hence there is no ambiguity.

2. Systems with pre defined, finite life.
   This might be the case of a transport module in a space mission. Once the payload has been placed in its final orbit, there is no advantage in prolonging operative life of the transport system. However, if we solve the problem above, we can design a system which reach its end life exactly when we want it. This systems would minimize cost.

To develop the thought above into a coherent and somewhat general theory, we will start a simple system as a guide model.

Hybrid car model hypothesis

As a guide system, we will use a hybrid car, comprising of two engines, one electric and one internal combustion.

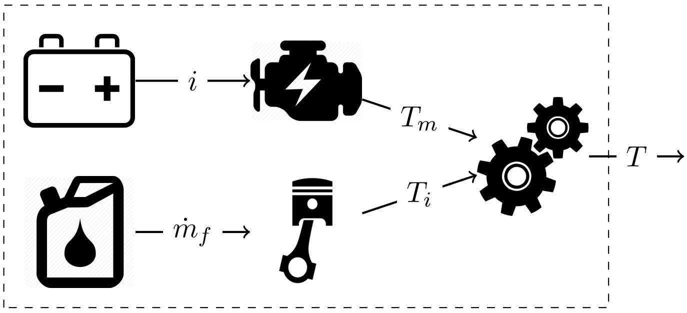


Figure 2.9: Our guide system for this chapter

Assuming **linear production functions**, the overall system output is given by

$$T = \alpha \cdot i + \beta \cdot \dot{m}_f \tag{2.3}$$

Each engine consumes a characteristic resource, current and fuel respectively. These are both finite resources, hence

$$\int_0^{t_f} i(t)\, dt = Ah \qquad \int_0^{t_f} \dot{m}_f(t)\, dt = M_{tot}$$

The goal is to last as long as possible, or to obtain t_f max. Formally, $T(t)$ being the torque requirement at a time t, we define

$$t_f \doteq \max \left\{ \hat{t} \in \mathbb{R}^+ \text{ such that } \int_0^{\hat{t}} i(t)\, dt \leq Ah \quad \int_0^{\hat{t}} \dot{m}_f(t)\, dt \leq M_{tot} \text{ and } \forall t \leq \hat{t},\, T(t) = \alpha i(t) + \beta \dot{m}_f(t) \right\}$$

The two functions $i(t)$ and $\dot{m}_f(t)$ are the variable we can use to maximize t_f.

Remark. Notation:

1. Ah is a measure of the battery capacity (Ampere hours).
2. In an attempt at a more intuitive notation, instead of using x_1 and x_2 to describe the operational level of the electric engine and the combustion engine, we will use i and $\dot{m}_f$. They are pure numbers in $[0, 1]$ and can be defined from the real measured consumption I and $\dot{M}$ as

$$i \doteq \frac{I}{I_{max}} \quad \left[\frac{\text{A}}{\text{A}}\right] \qquad \dot{m}_f \doteq \frac{\dot{M}}{\dot{M}_{max}} \quad \left[\frac{\text{kg}}{\text{kg}}\right]$$

How the cost function decides what to use

We have one equation in two unknowns. To solve the system, we need another equation, that is how much we value $\dot{m}_f$ compared to i. Let us add the generic linear cost function

$$\begin{cases} T = \alpha i + \beta \dot{m}_f \\ C = c_i \cdot i + c_m \cdot \dot{m}_f \end{cases}$$

Obviously, specific and total costs c_i, c_m, C are unknowns. The problem becomes finding a strategy to define c_1, c_2 so that minimizing the cost function will yield the longest t_f.

Form the second equation we obtain an equivalence among the two goods. The same cost can be paid by either using only i or $\dot{m}_f$.

$$C = c_i \cdot i = c_m \cdot \dot{m}_f \quad \Rightarrow \quad \dot{m}_f = \frac{c_i}{c_m} \cdot i = k \cdot i$$

The parameter k sets an equivalence between electric energy consumption and fuel consumption. It states the relative preference between the two.

Remark. Note that:

1. Only relative prices matter. If we have L goods, and $\vec{C} \in \mathbb{R}^L$ is a cost vector, the solution to the optimal management problem will be the same for every $\lambda \vec{C}, \lambda \in \mathbb{R}^{++}$ cost vectors. Since we only need to define $L - 1$ relative prices, in our case the parameter k is enough ($L - 1 = 2 - 1 = 1$).
2. Here, we are using i and $\dot{m}_f$ as quantities, against what we previously stated on them begin operational levels. However, under the hypothesis of *linear production functions*[7], the two differ by a mere constant, which we can imagine included in the costs.

[7] Even without linear production there are very tightly bounded.

Now, we minimize cost $C' \doteq \frac{C}{c_m}$ and see which k leads to maximum t_f.

$$\begin{cases} T = \alpha i + \beta \dot{m}_f \\ C' = k \cdot i + m_f \end{cases} \qquad T = \alpha i + \beta \cdot (C' - ki) \quad C' = \frac{T + (\beta k - \alpha) i}{\beta}$$

Since $\dot{m}_f = f(i)$, the only way we can minimize cost is by acting on i,

$$\frac{\partial C'}{\partial i} = \frac{(\beta k - \alpha)}{\beta}$$

The sign of the derivative is given by

$$\beta k > \alpha \quad \Leftrightarrow \quad \frac{\beta}{c_m} > \frac{\alpha}{c_i}$$

Notice that

- $\frac{\partial C'}{\partial i} > 0$;Increasing i we increase cost. The amount of torque produced by fuel that we can buy with one unit of currency is greater than that using electricity. The, by increasing i (hence decreasing $\dot{m}_f$) we increase cost. We chose i as low as possible, while still meeting the require T.
- $\frac{\partial C'}{\partial i} < 0$; increasing i we decrease cost. To minimize cost, we require i as big as possible, $i = 1$.

Discriminant cases

Every point on the plane $i, \dot{m}_f$ in fig (2.12) can be associated to a torque value. By tracing constant value lines for $T(t)$, we can identify some critical values.

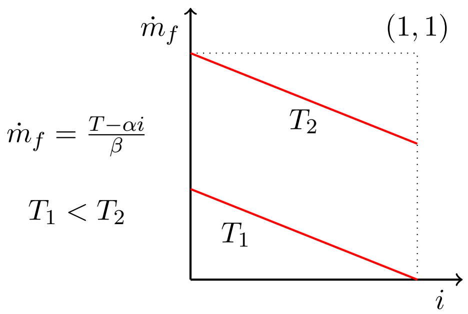


Figure 2.10: Production function $T(i, \dot{m}_f)$ on the operational level domain

We can identify two notable values for the output T. If the requested torque T is below a certain level ($T \leq T_1$), the level curves intersect both axis. This means that, even if one of the two subsystem were to shut down, we would be still able to meet the required T. While $T \leq T_1$, maximum consumption of i means zero consumption of $\dot{m}_f$

and vice versa.
For values above this threshold ($T_1 \leq T \leq T_2$) we can meet requirements with just the combustion engine, if we so chose. We can not do the opposite. Hence, even maximum consumption of i ($i = 1$) requires $\dot{m}_f \neq 0$ to produce T. For values $T \geq T_2$ we need to use both systems at all times.
We will call the first case **complete redundancy**, the second **partial redundancy** while the third **cooperative redundancy**.
Note that the discriminant among the 3 cases is quite simple

$$T_1 = \min\{\alpha, \beta\} \quad T_2 = \max\{\alpha, \beta\}$$

It is clear then that, if we run out of fuel, every request of T above T_1 will not be feasible, and the system will become non responsive. This is a simple case in which there are objective reasons to regard fuel as more valuable than current for this subsystem.

Constant parameters

We assume complete knowledge of all time varying constant $T(t)$ and that $\alpha(t) = \alpha, \beta(t) = \beta$ are both constant. We want to and find analytical equation for t_f, which we want to maximize, eventually using the parameter k.

Complete redundancy ($T \leq \min\{\alpha, \beta\}$)

To begin with, assume that T is constant; since both engine can meet requirements independently of each other, we do not need to worry about which is going to run out of resources first. With a simple integration we have that

$$\int_0^{t_f} T dt = \int_0^{t_f} \alpha i(t)\, dt + \int_0^{t_f} \beta \dot{m}_f(t)\, dt$$

$$T \cdot t_f = \alpha \cdot \int_0^{t_f} i(t)\, dt + \beta \cdot \int_0^{t_f} \dot{m}_f(t)\, dt = \alpha \cdot Ah + \beta \cdot M_{tot} \quad \Rightarrow \quad t_f = \frac{\alpha \cdot Ah + \beta \cdot M_{tot}}{T}$$

Which is clearly independent on $i(t), \dot{m}_f(t)$ and therefore k.

This result holds even if $T(t)$ is a generic function, but always less than T_1. Hence, since how we distribute resources doesn't matter, t_f **is independent of the price vector**.

Partial redundancy ($\min\{\alpha, \beta\} \leq T \leq \max\{\alpha, \beta\}$

Assume again that $T(t)$ is a constant; the case above applies with only minor modifications.

Example 2.15. Assume the electric engine is unable to provide the torque T alone. The order in which we do things does matter, but only slightly. If we consume all fuel first, we will not be able to use the electrical energy stored in the battery. Then, we run the electric engine with the internal combustion until the charge in the battery is depleted, hence we continue simply by burning fuel.

$$i_1, \dot{m}_{f1} \qquad i_2 = 0, \dot{m}_{f2} \qquad \text{such that } \alpha i_s + \beta \dot{m}_{f,s} = T$$

$$T \cdot t_f = \alpha \cdot \int_0^{t_f} i\, dt + \beta \cdot \int_0^{t_f} \dot{m}_f\, dt = \alpha \cdot \left(\int_0^{t_1} i_1\, dt + \int_{t_1}^{t_f} i_2\, dt \right) + \beta \cdot \left(\int_0^{t_1} \dot{m}_{f1}\, dt + \int_{t_1}^{t_f} \dot{m}_{f2}\, dt \right)$$

What if $T(t)$ varies with time? It still doesn't matter what we chose; any function $i(t)$ implies a fuel consumption function

$$\dot{m}_f(t) = \frac{T(t) - \alpha \cdot i(t)}{\beta}$$

which yield the same t_f. The only condition is that we consume all electric energy before we run out of fuel

$$t_i \leq t_m \quad \text{where } t_i : \int_0^{t_i} i(t)\, dt = Ah \quad \text{and } t_m : \int_0^{t_m} \dot{m}_f(t)\, dt = M_{tot}$$

If this holds, then $\forall i(t)$ we can define $\dot{m}_f$ as

$$\dot{m}_f \doteq \begin{cases} \frac{T(t) - \alpha \cdot i(t)}{\beta} & \text{for} \quad 0 \leq t \leq t_i \\ \\ \frac{T(t)}{\beta} & \text{for} \quad t_e < t \leq t_f \end{cases}$$

Cooperative redundancy ($\min\{\alpha, \beta\} \leq T \leq \alpha + \beta$)

Since both engines are needed to meet the requirement, they can not be run independently of one another. As soon as we run out of one resource, our system will become unresponsive. We can compute both *exhaustion* times t_i and t_m as

$$t_i \text{ such that } \int_0^{t_i} i(t)\, dt = Ah \qquad t_m \text{ such that } \int_0^{t_m} \dot{m}_f(t)\, dt = M_{tot}$$

If the request T is constant in time, we have good reasons to think that $i(t)$ and $\dot{m}_f(t)$ will be constant as well. We will verify this hypothesis at the end. Then we can write

$$\int_0^{t_i} i\, dt = i \cdot t_i = Ah \Rightarrow t_i = \frac{Ah}{i} \qquad \int_0^{t_m} \dot{m}_f\, dt = \dot{m}_f \cdot t_m = M_{tot} \Rightarrow t_m = \frac{M_{tot}}{\dot{m}_f}$$

And $t_f \doteq \min\{t_i, t_m\}$. Since i and $\dot{m}_f$ are linked by the required torque, we can write both depletion times as a function of i

$$t_i(i) = \frac{Ah}{i}; \qquad \dot{m}_f = \frac{T - \alpha i}{\beta} \Rightarrow t_m(i) = \frac{M_{tot} \cdot \beta}{T - \alpha i}$$

We plot $t_i(i)$ and $t_m(i)$

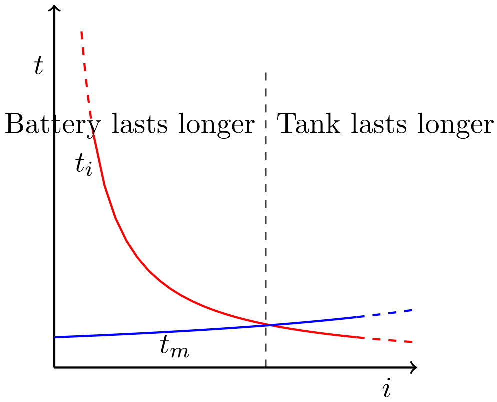


Figure 2.11: Depletion time as a function of current

t_f will be the greatest if we finish all resources[8] at the same time.

$$t_m(i) = t_b(i) \quad \Rightarrow \quad \frac{M_{tot} \cdot \beta}{T - \alpha i} = \frac{Ah}{i} \quad \Rightarrow \quad i = \frac{Ah \cdot T}{M_{tot} \cdot \beta + Ah \cdot \alpha}$$

And, not surprisingly, there is a clear symmetry

$$\dot{m}_f = \frac{T - \alpha i}{\beta} = \frac{T - \alpha \frac{Ah \cdot T}{M_{tot} \cdot \beta + Ah \cdot \alpha}}{\beta} = \frac{M_{tot} \cdot T}{M_{tot} \cdot \beta + Ah \cdot \alpha}$$

We can find the ratio of i to $\dot{m}_f$;

$$k = \frac{c_i}{c_m} = \frac{i}{\dot{m}_f} = \frac{Ah}{M_{tot}}$$

The line $\dot{m}_f = \frac{1}{k} \cdot i$ is the preferred relationship between i and $\dot{m}_f$, which is the relationship that maximizes t_f.

[8]Note that, for the way we have constructed this model, the two engine can only cooperate, they can not be detrimental to each other (ie one can not break while the other accelerates).

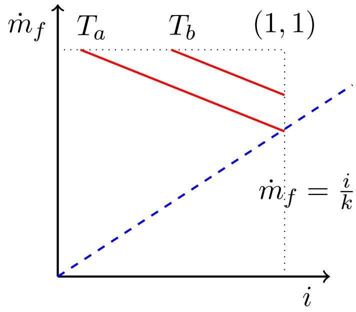


Figure 2.12: Required torque and maximum t_f strategy.

If the two curves intersect **within** the operational level domain (case T_a in figure 2.12), we are able to produce the right torque and achieve the condition of maximum t_f. The analytical condition for this to happen are the following

$$\begin{cases} T = \alpha \cdot i + \beta \cdot \dot{m}_f \\ \dot{m}_f = \frac{i}{k} \end{cases} \Rightarrow \quad T = \left(\alpha + \frac{\beta}{k}\right) \cdot i \quad i = \frac{T}{\alpha + \frac{\beta}{k}} \leq 1 \quad \text{and} \quad \dot{m}_f = \frac{i}{k} \leq 1$$

When this happens, **both resources are being used at the same rate**. This means that k is constant in time as well. Hence, the solution to the system above are time invariant, and our initial hypothesis is verified($i, \dot{m}_f$ are constant).

This results holds whenever we satisfy the equation $\dot{m}_f = \frac{i}{k}$. However, even if we decide to move on the line of equi-consumption we will consume fuel and current at the same time, but producing different levels of T.

If the two curves don't intersect (such as T_b figure 2.12), we have to compromise. To maximize t_f we try to get as close as possible to the optimal condition. To measure this distance, we can identify a family of lines parallel to the optimum,

$$\dot{m}_f = \frac{1}{k} \cdot i + C \qquad C \in \mathbb{R}$$

which naturally generates the cost function $C : \mathbb{R}^2 \to \mathbb{R}$ that conveys the idea of *distance from the optimum*

$$C = \dot{m}_f - \frac{1}{k} \cdot i$$

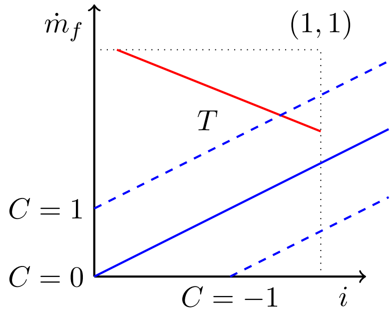


Figure 2.13: Required torque and cost function.

Let us study the sign of C

$$C : \begin{cases} \dot{m}_f - \frac{1}{k} \cdot i \geq 0 & \text{for} \quad \dot{m}_f \cdot k \geq i \\ \\ \dot{m}_f - \frac{1}{k} \cdot i \leq 0 & \text{for} \quad \dot{m}_f \cdot k \leq i \end{cases}$$

But, remembering that $k = \frac{Ah}{M_{tot}}$ and that $t_i = \frac{Ah}{i}$ and $t_m = \frac{M_{tot}}{\dot{m}_f}$

$$\dot{m}_f \cdot k \geq i \quad \Leftrightarrow \quad \dot{m}_f \cdot \frac{Ah}{M_{tot}} \geq i \quad \Leftrightarrow \quad \frac{Ah}{i} \geq \frac{M_{tot}}{\dot{m}_f} \quad \Leftrightarrow \quad t_i \geq t_m$$

Hence, positive value of C, mean that we deplete the fuel tank before the battery and vice versa.

In this instance, minimizing C will maximize t_f. We might start to question why we should go to all the trouble to define a cost function, when clearly we could just compute the responsive time directly. It is also clear that the two optimizations lead to the same result.

Remark. One immediate perk of not using the $t(f)$ that we could define over the whole domain is that such functions is somewhat dubious if either i and $\dot{m}_f$ are not constant. Without time constant assumption, we no longer have that

$$t_i = \frac{Ah}{i} \qquad t_m = \frac{M_{tot}}{\dot{m}_f}$$

By doing so, iso cost lines do not represent iso time line.

If $t_i \leq t_m \Rightarrow t_f = t_i = \frac{Ah}{i}$(case below the blue line). Then, all the points that share the same current consumption i, share the same t_f. In the same way for $t_m \leq t_m$, $t_f = \frac{M_{tot}}{\dot{m}_f}$. If we plot iso-time lines we have

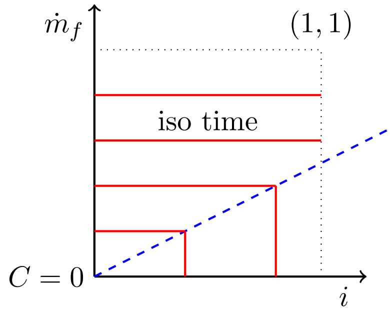


Figure 2.14: Iso responsive time curves(red) Vs cost (blue).

Generic $T(t)$

Let us now consider $T(t)$ as a generic function which can take any value in the admitted domain.

As we have seen before, whenever $T(t)$ is below the threshold of complementary redundancy, the choice among i and $\dot{m}_f$ isn't a meaningful one. That is, any choice will allow for the same responsive time. This is a key observation, because it gives us the possibility of action. What we chose in this region of request **will not affect our responsive time** *in this region* but will still have some effects.

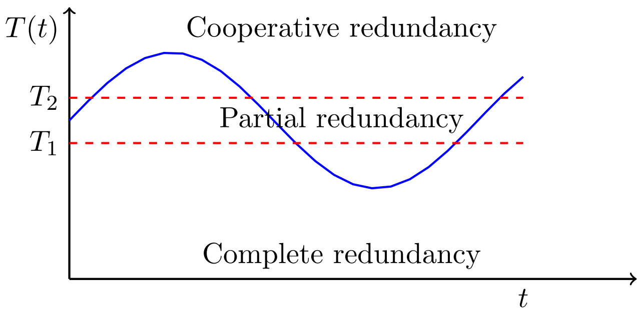


Figure 2.15: Generic torque $T(t)$

It is easy to see that the zero always belongs to the equi-consumption line; for sufficiently low level of output then, we will always be able to operate on this line. If we do so, we have seen that the value k will not change.
If we operate outside of this line, either because we have to (there is no other possibility, as $T(t') > T_2$) or because we decide to (if $T(t') < T_2$ we can pretty much do whatever we want) we will consume resources unevenly. Assume for example that we decide to consume more current than fuel; we will deplete the battery more than thee fuel tank and the ratio $k = \frac{Ah}{M_{tot}}$ will decrease.
Hence, by deciding the operation level, we can modify the shape of the equi-consumption

line. This possibility opens a new degree of freedom, which allows us to devise a general strategy.

It is clear that, among all of the possible k, the best is that is $k = 1$. This way, we can provide any torque level without ever leaving the equi-consumption line. This means that any function $T(t)$ will be met and responsive time will be maximized.

$k = 1$ means that we have the same amount[9] of resources. If this is the preferred condition, it is clear that given any resource level $Ah(t), M_{tot}(t)$, we will always try to consume more of the most abundant resource. This is usually a viable option only when $T(t) < T_2 = \max(\alpha, \beta)$.

We have therefore a simple algorithm that maximizes responsive life.

Response time maximizing Algorithm

labelprotocol1

if $k \neq 1$ **then**
    if $T(t) \geq T_2$ **then**
        Minimize $|C|$
    else
        Consume most abundant resource to $k \to 1$
    end if
else
    $\dot{m}_f = i \Rightarrow i(t) = \frac{T(t)}{\alpha+\beta}$
end if

Remark. Note that

1. This algorithm **does not require prior knowledge** of $T(t)$ as it operates on an instantaneous fashion.

2. It will give rise to non continuous control functions; some adjustment can be made, but it will shorten responsive life.

Upper Bound

It is a good idea to find a criterion to measure how good any proposed solution might be. The maximum possible time t_f will occur if the two engine can operate independently, the maximum torque we can offer is given by $\alpha \cdot Ah + \beta M_{tot}$. Then

$$t_{f,max} \doteq t \text{ such that } \int_0^t T(t)dt = \alpha \cdot Ah + \beta M_{tot}$$

Remark. The time t_f we obtain from the above is not necessarily an achievable maximum. We can not however find a higher value.

[9] Two quantities here are equal if they can produce the same torque output

An application: 3 reaction wheels for 2D satellite manoeuvre

We want to increase the complexity of the problem by increasing the number of dimensions. This will help us define and understand the basic concept which are needed to reach general results.

Assume we have a 2D model of a satellites that uses reaction wheels to turn around 2 axis. For redundancy it is equipped with 3 reaction wheels; one on the x positive axis, one on the positive y axis, and the third on the diagonal.

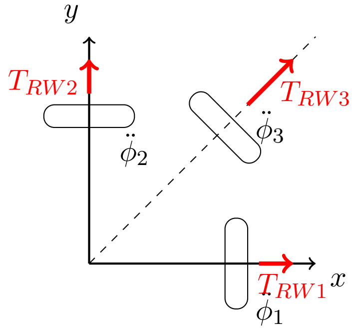


Figure 2.16: Model of a satellite that can rotate in x and y

$$\begin{cases} T_x &= T_{RW1} &+ T_{RW3} \cdot \sin(\theta) \\ T_y &= T_{RW2} &+ T_{RW3} \cdot \cos(\theta) \end{cases} \Rightarrow \begin{cases} T_x &= I_1 \cdot \ddot{\phi}_1 &+ \frac{I_3}{\sqrt{2}} \cdot \ddot{\phi}_3 \\ T_y &= I_2 \cdot \ddot{\phi}_2 &+ \frac{I_3}{\sqrt{2}} \cdot \ddot{\phi}_3 \end{cases} \tag{2.4}$$

Each reaction wheels consumes current, however this is a homogeneous resource, and we already know how to deal with it. Instead, will consider it's rotational velocity/degree of saturation as the limiting condition. This gives us a condition on maximum velocity

$$\dot{\phi}_1 \leq \dot{\phi}_{1,max} \quad \dot{\phi}_2 \leq \dot{\phi}_{2,max} \quad \dot{\phi}_3 \leq \dot{\phi}_{3,max}$$

The goal then is to maximize responsive time, or the interval between desaturation manoeuvres. Since the operational level will be the angular acceleration $\ddot{\phi}$, the saturation condition can be expressed by

$$\int_0^{t_f} \ddot{\phi}_i(t)\, dt \leq \dot{\phi}_{i,max} \doteq I_i \qquad i = 1, 2, 3$$

Possibly to cause more confusing, the maximum rotational velocity will be identified by I_i.

Remark. Note that;

1. In this application, maximizing responsive time means maximizing time between desaturation periods. This has the advantage of reducing *down time* in the satellite pointing capabilities. Furthermore, if the disturb was periodic, one might be able to avoid de-saturation all together, if one were able last more than half period.

2. In this example, we will assume that $T_x(t), T_y(t) \geq 0$ for every t. A more generic example will be discussed later however, this should be the worst case scenario.

3. To begin with, we assume no *explicit* transferring of angular momentum from one reaction wheel to the other. This means asking that $\ddot{\phi}_i \geq 0$ at all times.
   This restriction conceptually treats saturation of the wheels as any other resources (which can not be traded for one another). Later it will become clear that, by lifting this restriction, nothing changes.

In order to avoid any assumption on geometry and reaction wheel parameters, we rewrite equation as

$$\begin{cases} T_x &= \alpha \cdot \ddot{\phi}_1 + \gamma_1 \cdot \ddot{\phi}_3 \\ T_y &= \beta \cdot \ddot{\phi}_2 + \gamma_2 \cdot \ddot{\phi}_3 \end{cases} \Rightarrow \begin{pmatrix} T_x \\ T_y \end{pmatrix} = \begin{bmatrix} \alpha & 0 & \gamma_1 \\ 0 & \beta & \gamma_2 \end{bmatrix} \cdot \begin{pmatrix} \ddot{\phi}_1 \\ \ddot{\phi}_2 \\ \ddot{\phi}_3 \end{pmatrix}$$

Having 3 variables and 2 equations, we have one degree of freedom; a generic solution can be written as

$$\begin{pmatrix} T_x \\ T_y \end{pmatrix} = \begin{bmatrix} \alpha & 0 & \gamma_1 \\ 0 & \beta & \gamma_2 \end{bmatrix} \cdot \left(\begin{pmatrix} \frac{T_x}{\alpha} \\ \frac{T_y}{\beta} \\ 0 \end{pmatrix} + \begin{pmatrix} -\frac{\gamma_1}{\alpha} \\ -\frac{\gamma_2}{\beta} \\ 1 \end{pmatrix} \ddot{\phi}_3 \right) \Rightarrow \vec{T} = [\mathbf{B}](\vec{X} + \vec{K} \cdot \ddot{\phi}_3) \qquad \begin{array}{ll} \vec{K} & \in \mathrm{Ker}([\mathbf{B}]) \\ \vec{X} & \text{such that } \vec{T} = [\mathbf{B}] \cdot \vec{X} \end{array}$$

Since we have only one degree of freedom but two independent outputs, in some *yet loose sense*, we have less choices than in the previous example (in which the dimension of the output and degree or redundancy were equal).

Following the same approach as before, we want to define the regions of complete, partial and cooperative redundancy. This will allow us to define the co domain of the $\vec{T}$ that we can hope to provide given any level of saturation.

We need to be careful in extending the previous definition. Before, the complete redundancy set was comprised of all the requests to which every subsystem was able to answer independently. This is no longer a meaningful distinction since, in general, a single subsystem can only supply a mono dimensional output, while the request is typically bi dimensional.

To derive a more insightful generalization, let us plot all the possible $\vec{T}$ on a plane (figure 2.17).

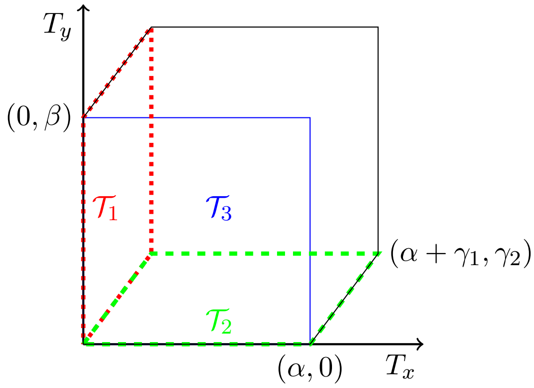


Figure 2.17: All possible output levels

The highlighted sets $\mathcal{T}_i$ are the sets of $\vec{T} \in \mathbb{R}^2$ that don't require the i-th reaction wheel. The intersection of the three $\mathcal{T}_{1,2,3} = \mathcal{T}_1 \cap \mathcal{T}_2 \cap \mathcal{T}_3$ can be viewed as the **complete redundancy set**. In fact, for all $\vec{T} \in \mathcal{T}_{1,2,3}$, we can take out any subsystem and still be able to meet $\vec{T}$.

Using those same elementary sets, we can define the **partial redundancy sets** defined as those $\vec{T}$ that we can not supply unless we have two specific subsystems; these are

$$\mathcal{T}_1 \setminus \mathcal{T}_{1,2,3} \qquad \mathcal{T}_2 \setminus \mathcal{T}_{1,2,3} \qquad \mathcal{T}_3 \setminus \mathcal{T}_{1,2,3}$$

Finally, we have the set of $\vec{T}$ that require all 3 reaction wheels to be operational[10], which will be the **cooperative redundancy set**.

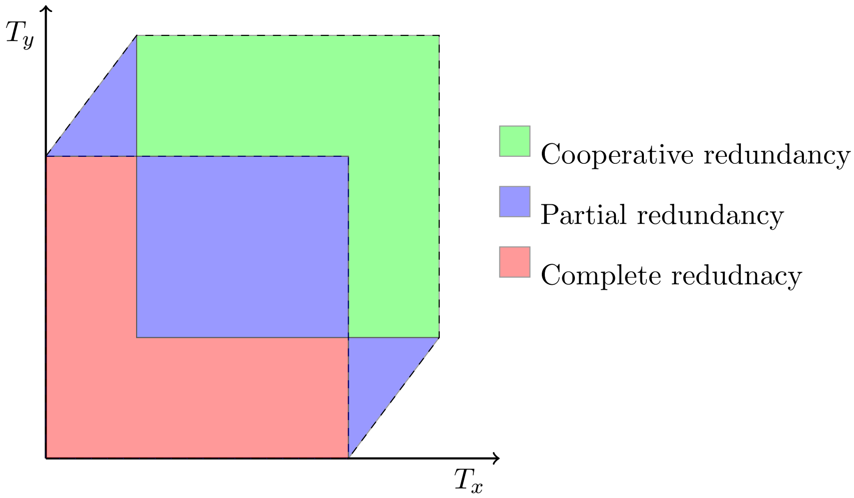


Figure 2.18: Complete, Partial and Cooperative redundancy

[10]This is something more than the inclusion of all the previous sets.

Ideally, we would like to saturate all reaction wheels in order to get the maximum responsive time. We have 3 choices;

1. We saturate one reaction wheel, and then the other two at the same time.

2. We saturate two reaction wheels, and then the remaining one.

3. We saturate all three at the same time.

Both option 1 and 2 are viable only under specific conditions on $\vec{T}$. The first is generally viable only if $\vec{T} \in \mathcal{T}_{1,2,3}$ (otherwise, as soon as we lose one reaction wheel, we become un-responsive). Option 2 is usually not feasible, unless $\vec{T}$ is parallel to $(\alpha, 0)^T$, $(0, \beta)^T$ or $(\gamma_1, \gamma_2)^T$.

Complete redundancy

Assume $\vec{T}(t) = \vec{T}$ constant and $\vec{T} \in \mathcal{T}_{1,2,3}$. This means that we could saturate any of the reaction wheels and still be able to meet requirements. However, once we have only 2 reaction wheels, the system becomes completely determined; we lose the ability to chose anything. Therefore, the residue responsive time is determined only by $\vec{T}(t)$. To maximize responsive time, we try to saturate all reaction wheels at the same time.

Saturating one RW

This strategy relies too heavily on the assumption of constant T, and therefore will not be pursued.

Saturating every reaction wheel at the same time

This strategy is by far the best; it can also be applied to partial and complete redundancy.

We compute residue time under the hypothesis of constant $\ddot{\phi}_i(t)$. We have to equations in 3 variables, therefore everything can be expressed as a function of $\ddot{\phi}_3$.

$$t_{RWi} = \frac{I_i}{\ddot{\phi}_i} \qquad \begin{cases} T_x = \alpha \cdot \ddot{\phi}_1 + \gamma_1 \cdot \ddot{\phi}_3 \\ T_y = \beta \cdot \ddot{\phi}_2 + \gamma_2 \cdot \ddot{\phi}_3 \end{cases} \Rightarrow t_{RW1} = \frac{I_1 \cdot \alpha}{T_x - \gamma_1 \phi_3} \quad t_{RW2} = \frac{I_2 \cdot \beta}{T_y - \gamma_2 \phi_3}$$

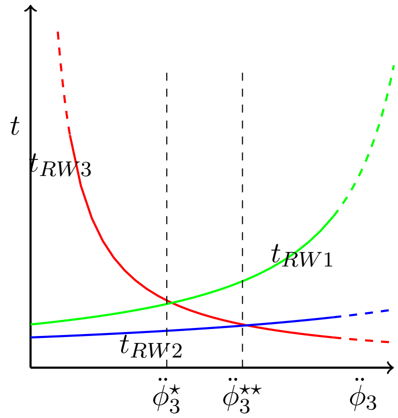


Figure 2.19: Depletion time as a function of RW3 operation

Since we are don't want to lose our ability to chose, we restrict responsive time as $t_f \doteq \min\left\{t_{RW1}(\ddot{\phi}_3), t_{RW2}(\ddot{\phi}_3), t_{RW3}(\ddot{\phi}_3)\right\}$. Given the fact that the curves $t_{RWi}(\ddot{\phi}_3)$ have opposite derivatives[11] the maximum for t_f will be when two or more intersect.

Theoretically, there can be three points in which we saturate two reaction wheels at the same time; this are given by

$$t_{RW1} = t_{RW2} \doteq t_{1,2} \Rightarrow \frac{I_1 \cdot \alpha}{T_x - \gamma_1 \phi_3} = \frac{I_2 \cdot \beta}{T_y - \gamma_2 \phi_3} \quad t_{1,3} \Rightarrow \frac{I_1 \cdot \alpha}{T_x - \gamma_1 \phi_3} = \frac{I_3}{\phi_3} \quad t_{2,3} \Rightarrow \frac{I_2 \cdot \beta}{T_y - \gamma_2 \phi_3} = \frac{I_3}{\phi_3}$$

Which give the three operational levels

$$t_{1,2} \Rightarrow \ddot{\phi}_3^{\star} = \frac{I_1 \alpha T_y - I_2 \beta T_x}{I_1 \alpha \gamma_2 - I_2 \beta \gamma_1} \qquad t_{1,3} \Rightarrow \ddot{\phi}_3^{\star\star} = \frac{I_3 T_x}{I_1 \alpha + I_3 \gamma_1} \qquad t_{2,3} \Rightarrow \ddot{\phi}_3^{\star\star\star} = \frac{I_3 T_y}{I_2 \beta + I_3 \gamma_2}$$

For each, we must check whether we saturate first the coupled reaction wheels or the third one:

$$t^{\star} = \min\left\{t_{1,2}, t_{RW3}(\ddot{\phi}_3^{\star})\right\} \qquad t^{\star\star} = \min\left\{t_{1,3}, t_{RW2}(\ddot{\phi}_3^{\star\star})\right\} \qquad t^{\star\star\star} = \min\left\{t_{2,3}, t_{RW1}(\ddot{\phi}_3^{\star\star\star})\right\}$$

And finally, we will chose $\ddot{\phi}_3$ that yield the maximum t_f.

Generally then, **we can not saturate all reaction wheels at the same time**. If $\ddot{\phi}_3^{\star} \neq \ddot{\phi}_3^{\star\star}$, there is nothing we can do to change it. We can only chose the one that yields the longest time.

[11] This is intuitive as they are derived under the constraint of equal output. If one reaction wheels contributes less to the torque output, another will have to supply the difference.

If $\ddot{\phi}_3^\star = \ddot{\phi}_3^{\star\star}$ we are able to saturate all reaction wheels at the same time, hence maximum responsive time would be achieved. However, as noted before, this condition might not be verified, depending both on $\vec{T}$ and I_1, I_2, I_3. By equating the two we obtain

$$\frac{I_1\alpha + I_3\gamma_1}{T_x} = \frac{I_2\beta + I_3\gamma_2}{T_y} \quad \Rightarrow \quad \alpha T_y \cdot I_1 - \beta T_x \cdot I_2 + (T_y\gamma_1 - T_x\gamma_2) \cdot I_3 = 0$$

The best strategy to adopt is then determined by relationships between I_1, I_2, I_3 and T_x, T_y. In the hope of a more intuitive and formal expression of these conditions, we introduce the *resource space.*

Interpretation using the space of resources

Consider the resource space I_1, I_2, I_3 as an euclidean vector space. The resource currently available are the points of this space, while the vectors are the instantaneous consumption $\vec{\Phi} = (\ddot{\phi}_1, \ddot{\phi}_2, \ddot{\phi}_3)^T \mathrm{dt}$. In this space, a request $\vec{T}$ generates a plane of equi consumption possibilities, π, given by

$$\pi: \quad \begin{bmatrix} \alpha T_y & -\beta T_x & (T_y\gamma_1 - T_x\gamma_2) \end{bmatrix} \cdot \begin{pmatrix} I_1 \\ I_2 \\ I_3 \end{pmatrix} = 0 \tag{2.5}$$

If we are on this plane, we can move on a straight line towards the origin (which is kind of our nirvana, the condition in which we deplete all resources and therefore reach t_f maximum). Each point on this plane is associated with a maximum t_f strategy, given by :

$$\vec{\Phi}^\star = \begin{pmatrix} \frac{T_x - \gamma_1\ddot{\phi}_3}{\alpha} \\ \frac{T_y - \gamma_2\ddot{\phi}_3}{\beta} \\ \ddot{\phi}_3 \end{pmatrix} \qquad \ddot{\phi}_3 = \frac{I_3 T_x}{I_1\alpha + I_3\gamma_1} = \frac{I_3 T_y}{I_2\beta + I_3\gamma_2} \tag{2.6}$$

Notably, we can only move parallel to the plane π. This is easily shown by remembering that the allowed motions are

$$\vec{\Phi} = \begin{pmatrix} \frac{T_x}{\alpha} \\ \frac{T_y}{\beta} \\ 0 \end{pmatrix} + \begin{pmatrix} -\frac{\gamma_1}{\alpha} \\ -\frac{\gamma_2}{\beta} \\ 1 \end{pmatrix} \ddot{\phi}_3 \qquad \begin{bmatrix} \alpha T_y & -\beta T_x & (T_y\gamma_1 - T_x\gamma_2) \end{bmatrix} \cdot \vec{\Phi} = 0$$

Although this shouldn't come as a surprise, it has some notable implications.

Remark. This means that

1. If we are on π, and the requirements don't change, we can move *freely* on the plane and still be able to revert back to $\ddot{\phi}_3^\star$ to finish all resources at the same time.

2. We can not really move *freely* on the plane, as we have 2 equations in 3 unknowns. Wherever we are on the plane, we can still reach the origin.

3. Since we move parallel to this plane, if we start outside of it (and it doesn't change), we won't be able to reach it.

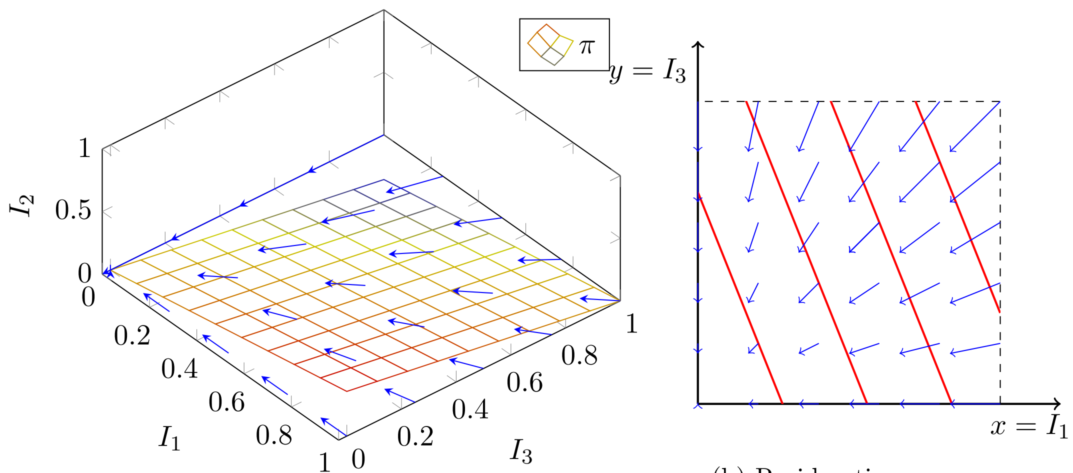
If we are on the plane however, we can make sense of some intuitive results; since the *optimal path* $\vec{\Phi}^\star$ on the plane is defined as

$$\vec{\Phi}^\star = \begin{pmatrix} \frac{T_x I_1}{I_1\alpha + I_3\gamma_1} \\ \\ \frac{T_y I_2}{I_2\beta + I_3\gamma_2} \\ \\ \frac{T_x I_3}{I_1\alpha + I_3\gamma_1} \end{pmatrix}$$

And it's direction is straight to the origin. To see that, one can check

$$\vec{I}_{t0} + s\vec{\Phi}^\star = 0 \quad \text{for some } s \in \mathbb{R}$$

In fact $s = \frac{I_3}{\phi_3}$. Once again, this renders the idea that, if something is scarce ($I_1 \approx 0$) we will value it more (and move almost parallel to I_1). We have formally expressed what we always knew, we need to move straight towards the origin.



(a) Plane π.

(b) Residue time on π

Analytical results

In the previous section, we have worked under the simplifying assumption of $\vec{T}(t)$ constant. This is an extremely strong assumption, but its usefulness lays in introducing intuitive concepts and guidelines for a more general and useful approach. Our final goal for this branch of management will be to develop a *management strategy* able to maximize responsive time against an unknown $\vec{T}(t)$.
The next step towards this goal is to assume $\vec{T}(t)$ generic, but known.

As $\vec{T}$ is the key factor that distinguishes the various cases, we begin by establishing its basic proprieties. $\vec{T}(t)$ represents the vector of requests at any time t; each component is a request for the system to produce something. We assume that, to meet the requirement T_i the system will **always consume resources** and **never gain them**.

This assumption is justified by the fact that, in most cases, subsystem do not show a symmetrical behaviour; a combustion engine does not produce gasoline when a torque is applied, and an electric engine has not the same proprieties when used as a motor or a generator. However, we will challenge this hypothesis by the end of the chapter, as resupply considerations are crucial for an autonomous system.
Due to the sign convention, to maintain the same sing on the resource consumption, we assume that every component of $\vec{T}$ has to be greater or equal to 0 at any time. This is expressed by

$$T_i(t) \geq 0 \quad i = 1, 2, ...n \, \forall t \in \mathbb{R}^+ \qquad \Leftrightarrow \qquad \vec{0} \ll \vec{T}(t) \, \forall t \in \mathbb{R}^+ \tag{2.7}$$

Since we intend to use responsive time to measure the effectiveness of our choices we need to produce an unambiguous definition. Intuitively, responsive time is the period from an arbitrarily set starting point and the last instant in which the system is able to meet demand. It is clear, however, that is the request is constantly $\vec{0}$ any algorithm is able to meet demand indefinitely. Whenever there is an interval $[t_1, t_2]$ in which $\vec{T}(t) = \vec{0}$, ambiguity arises. To avoid this we simply impose

$$\text{if } \vec{T}(t') = \vec{0} \text{ Then } \exists \delta \text{ such that } \forall \, \varepsilon < \delta \in \mathbb{R}^+ \quad \vec{T}(t' + \varepsilon) \neq \vec{0}, \tag{2.8}$$

This only means we have to *pause* our time counter whenever the system goes through a period of hibernation.

We want to obtain some formal results to guide our algorithm. It is easy to observe that our model of system behaviour is *conservative in a broad sense.* Every unit of resources can be directly transformed into a given amount of output, and the conversion has a fixed and constant[12] efficiency. It is therefore reasonable to think that *how we use it won't affect how much output we can obtain.* This builds on the hypothesis of linearity and time constant matrix $\mathbf{B}$ with all non negative elements[13].

[12]Fixed as linear with production, constant as constant in time.

[13]If there are positive and negative elements, it means that some components are consuming what other are producing, which is a case we will not consider here

Hence, we can reasonably expect that **using more resources will yield a longer responsive time**. As of now, *using more resources* is a rather loosely defined concept, since it would requires a total order in a subspace of $\mathbb{R}^n$. However, if we consume **all available resources**, we are sure to have used *more resources* than in any other case. We will naturally start by aiming at the origin of the resources space. Many question arise naturally, such as, is it always possible to consume al resources? Can we prove that it doesn't matter how we consume the initial resources?
We will start from the latter question.

Theorem 1. Let $\vec{T}(t) : \mathbb{R}^+ \to \mathbb{R}^{n+}$ be a known demand function and $\vec{\Phi}(t) : \mathbb{R}^+ \to \mathbb{R}^m$ an instantaneous management function such that

$$\vec{T}(t) = \mathbf{B} \cdot \vec{\Phi}(t) \quad \forall t \in \mathbb{R}^+ \quad \text{ and } \mathbf{B} \in \mathbb{R}^{n\times m} \tag{2.9}$$

Then, if exists t_f such that,

$$\forall t \in [0, t_f) \quad \int_0^t \vec{\Phi}(t)\text{dt} \ll \vec{I}(0) \quad \text{and} \quad \int_0^{t_f} \vec{\Phi}(t)\text{dt} = \vec{I}(0) \tag{2.10}$$

t_f **is unique**.

This results tells us that *how we use the resources*, hence which $\vec{\Phi}(t)$ we chose, won't matter as long as we use all resources. Note that the dis-equivalence $\ll$ means for every component of a vector.

Proof. Since $\vec{\Phi}(t)$ has to meet demand and we assume t_f exists, we can integrate equation 2.9 from 0 to t_f

$$\int_0^{t_f} \vec{T}(t)\text{dt} = \int_0^{t_f} \mathbf{B} \cdot \vec{\Phi}(t)\text{dt} = \mathbf{B} \cdot \int_0^{t_f} \vec{\Phi}(t)\text{dt} = \mathbf{B} \cdot \vec{I}(0) \quad \Rightarrow \quad \int_0^{t_f} \vec{T}(t)\text{dt} = \mathbf{B}\vec{I}(0) \tag{2.11}$$

This is clearly independent of our choice of $\vec{\Phi}(t)$. Assume that t_f is not unique and it exists $t'_f \in \mathbb{R}^+ \setminus \{t_f\}$ that satisfies condition 2.10. Then, using equation 2.11

$$\int_0^{t'_f} \vec{T}(t)\text{dt} = \mathbf{B} \int_0^{t'_f} \vec{\Phi}(t)\text{dt} = \mathbf{B}\vec{I}(0) = \int_0^{t_f} \vec{T}(t)\text{dt}$$

Assume $t'_f > t_f$

$$\int_0^{t'_f} \vec{T}(t)\text{dt} = \int_0^{t_f} \vec{T}(t)\text{dt} + \int_{t_f}^{t'_f} \vec{T}(t)\text{dt} = \int_0^{t_f} \vec{T}(t)\text{dt} \quad \Rightarrow \quad \int_{t_f}^{t'_f} T(t)\text{dt} = 0$$

Since $T(t) \gg \vec{0}$, we would have to conclude $\vec{T}(t) = \vec{0}, \forall t \in [t_f, t'_f]$ but this is absurd due to hypothesis 2.8. $\square$

Form the proof of the theorem we can also obtain a condition for the existence of t_f. Assuming $T(t)$ known, we can determine whether we will be able to deplete all resources or not.

Example 2.16. Assume, as in the previous example, $\vec{\Phi} = (x_1, x_2, x_3)^T$, $\vec{T} = (T_x, T_y)^T$ and $\vec{I} = (I_1, I_2, I_3)^T$. System architecture is given by

$$\begin{pmatrix} T_x \\ T_y \end{pmatrix} = \begin{bmatrix} \alpha_1 & \beta_1 & \gamma_1 \\ \alpha_2 & \beta_2 & \gamma_2 \\ \alpha_3 & \beta_3 & \gamma_3 \end{bmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad \Rightarrow \quad \vec{T} = \begin{bmatrix} \vec{\alpha} & \vec{\beta} & \vec{\gamma} \end{bmatrix} \cdot \vec{X} \quad \Rightarrow \quad \vec{T} = \mathbf{B} \cdot \vec{X}$$

We want to know if t_f exists, or rather, which level of resources are able to allow for t_f to exists. Since there is one degree of redundancy, we can express everything as a function of one variable, say x_1. Hence,

$$\vec{T} - \vec{\alpha} \cdot x_1 = \begin{bmatrix} \vec{\beta} & \vec{\gamma} \end{bmatrix} \cdot \begin{pmatrix} x_2 \\ x_3 \end{pmatrix} \quad \Rightarrow \quad \begin{bmatrix} \vec{\beta} & \vec{\gamma} \end{bmatrix}^{-1} \int_0^{t_f} (\vec{T} - \vec{\alpha} \cdot x_1) \text{ dt} = \int_0^{t_f} \begin{pmatrix} x_2 \\ x_3 \end{pmatrix} \text{dt}$$

$$\begin{bmatrix} \vec{\beta} & \vec{\gamma} \end{bmatrix}^{-1} \left(\int_0^{t_f} \vec{T} \text{ dt} - \vec{\alpha} \cdot I_1(0) \right) = \begin{pmatrix} I_2(0) \\ I_3(0) \end{pmatrix} \quad \begin{pmatrix} I_1(0) \\ I_2(0) \\ I_3(0) \end{pmatrix} = \begin{pmatrix} I_1(0) \\ \begin{bmatrix} \vec{\beta} & \vec{\gamma} \end{bmatrix}^{-1} \left(\int_0^{t_f} \vec{T} \text{ dt} - \vec{\alpha} \cdot I_1(0) \right) \end{pmatrix}$$

$$\begin{pmatrix} I_1(0) \\ I_2(0) \\ I_3(0) \end{pmatrix} = \begin{pmatrix} 0 \\ \begin{bmatrix} \vec{\beta} & \vec{\gamma} \end{bmatrix}^{-1} \int_0^{t_f} \vec{T} \text{dt} \end{pmatrix} + \begin{pmatrix} 1 \\ -\begin{bmatrix} \vec{\beta} & \vec{\gamma} \end{bmatrix}^{-1} \vec{\alpha} \end{pmatrix} \cdot I_1(0) \tag{2.12}$$

Clearly, there is no reason why I_1 should play a predominant role, hence we expect similar results if we use any other variable x_i. We can also notice that the vector that multiplies I_1 is in the the ker of $\mathbf{B}$.

$$\mathbf{B} \cdot \begin{pmatrix} 1 \\ -\begin{bmatrix} \vec{\beta} & \vec{\gamma} \end{bmatrix}^{-1} \vec{\alpha} \end{pmatrix} = \begin{bmatrix} \vec{\alpha} & \vec{\beta} & \vec{\gamma} \end{bmatrix} \cdot \begin{pmatrix} 1 \\ -\begin{bmatrix} \vec{\beta} & \vec{\gamma} \end{bmatrix}^{-1} \vec{\alpha} \end{pmatrix} = \vec{\alpha} - \begin{bmatrix} \vec{\beta} & \vec{\gamma} \end{bmatrix} \cdot \begin{bmatrix} \vec{\beta} & \vec{\gamma} \end{bmatrix}^{-1} \cdot \vec{\alpha} = 0$$

Equation 2.12 gives, for every t_f, the set of initial resource that allow for the existence of t_f.

In general then, we can say that t_f exists when the initial condition satisfy 2.12.

Lemma 2. Let $\vec{T}(t) : \mathbb{R}^+ \to \mathbb{R}^{n+}$ be a known demand function for a system with m subsystem and m resources, described by a matrix $\mathbf{B}.t_f$ exists if $\vec{I}(t=0)$ is in the hyperplane of dimension $m - n$ that contains the point

$$\vec{P} = \begin{pmatrix} 0 \\ 0 \\ \dots \\ 0 \\ \\ \left[\mathbf{B}_{n \times n}\right]^{-1} \int_0^{t_f} T \text{dt} \end{pmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} 0 \\ 0 \\ \dots \\ 0 \end{matrix}}\right\} m-n \\ \left.\vphantom{\int}\right\} n \end{matrix}$$

and is generated by the null space of $\mathbf{B}$.

If condition 2.12 is not met, then it is not possible to end all resources at the same time. This will lead to some unusable *left overs*, or a final position $\vec{I}(t_f)$ different from the origin. Since we still want to find the path $\vec{\Phi}(t)$ that produces the longest responsive time, we need a function that associates a responsive time to each $\vec{\Phi}(t)$. As we had previously assumed however, we only need the initial and final point.

Lemma 3. Responsive time for a given $\vec{\Phi}(t)$ is a function only of $\vec{I}(0)$, $\vec{I}(t_f)$ and the demand function $\vec{T}(t)$.

Proof. Assume $\vec{I}(0) = (I_1, I_2, I_3)^T$ such that

$$\begin{bmatrix}\vec{\beta} & \vec{\gamma}\end{bmatrix}^{-1}\left(\int_0^{t_f} \vec{T}\ \mathrm{dt} - \vec{\alpha}\cdot I_1(0)\right) = \begin{pmatrix} I_2^\star \\ I_3^\star \end{pmatrix} \neq \begin{pmatrix} I_2 \\ I_3 \end{pmatrix} \quad \text{and, in particular} \quad \begin{cases} I_2^\star < I_2 \\ I_3^\star < I_3 \end{cases}$$

If we redefine the initial level of resources $\vec{I}$ as $(I_1, I_2^\star, I_3^\star)$, we have identically verified the condition of existence of t_f and from theorem 1, t_f has to be unique. Our final position will be $\vec{I}(t_f) = (0, I_2 - I_2^\star, I_3 - I_3^\star)^T$. Then t_f depends only on the $\vec{I}(0)$, $\vec{I}(t_f)$ and $\vec{T}(t)$. □

We must acknowledge the fact that $(I_2(0)^\star, I_3(0)^\star)^T$ might be $\ll \vec{0}$. To find the *appropriate* combination resources (one that is strictly greater than zero), we can add or subtract multiples of the null space until $\vec{I}^\star(0) \ll \vec{I}(0)$.

At this point, given initial and final position we can determine the elapsed responsive time **without having to specify the path we intend to chose**. The next step is to map all possible end-points in order to find which yields the highest responsive time.

Using lemma 3 it is clear that the time it take to go from A to C is the sum of times from A to B and B to B. Then we can arbitrarily set a starting point in the space resource I_0, define all responsive time from this point, and then, depending on the actual current location, redefine the t_f field over every point with a simple addition or subtraction.

Calculating the field of responsive time over the resource space.
Using the results from lemma 3, we can assign to *some specific points* in the space of resources a value t_f. The fact that not all point in the resource space are reachable should be clear from Lemma 2, which states whether is possible or not to reach the origin.
Then, our goal is to find the feasible set, or all the reachable points, and assign to each one a value t_f.
Given a point $\vec{0} \ll \vec{I'} \ll \vec{I}(0)$, it will be reachable if, for some $t^\star$ we have that the *leftover* when starting from $\vec{I}(0)$ is exactly that point. This can be written as

$$\Delta\vec{I} \doteq (\vec{I}(0) - \vec{I'})) \quad \vec{I}(0) = \Delta\vec{I} + \vec{I'} \quad \begin{bmatrix}\vec{\beta} & \vec{\gamma}\end{bmatrix}^{-1}\left(\int_0^{t^\star} \vec{T}\ \mathrm{dt} - \vec{\alpha}\cdot \Delta I_1(0)\right) = \begin{pmatrix} \Delta I_2(0)^\star \\ \Delta I_3(0)^\star \end{pmatrix}$$

We can notice two kinds of degrees of freedom, one accessed changing time t and the other changing the position of the end point, by acting on ΔI_1. We can easily draw

equi-responsive time sets by keeping $t^\star$ fixed and changing ΔI_1

$$I(t') = \left\{ I' = \vec{I}(0) - \Delta\vec{I} \quad \Delta\vec{I} = \begin{pmatrix} \Delta I_1 \\ \begin{bmatrix}\vec{\beta} & \vec{\gamma}\end{bmatrix}^{-1} \left(\int_0^{t'} \vec{T}\ \mathrm{dt} - \vec{\alpha}\cdot\Delta I_1\right) \end{pmatrix} \text{ for } \Delta I_1 \in [0, I_1(0)] \right\} \tag{2.13}$$

Once again, we recognise this set as a line, which we decompose as a point and a direction

$$I(t') = \vec{P} + \vec{K}\cdot s = \begin{pmatrix} I_1(0) \\ I_2(0) - [\vec{\beta}\,\vec{\gamma}]^{-1}\int_0^{t'} T_x\, dt \\ I_3(0) - [\vec{\beta}\,\vec{\gamma}]^{-1}\int_0^{t'} T_y\, dt \end{pmatrix} + \begin{pmatrix} -1 \\ [\vec{\beta}\,\vec{\gamma}]^{-1}\alpha_1 \\ [\vec{\beta}\,\vec{\gamma}]^{-1}\alpha_2 \end{pmatrix} \cdot \Delta I_1$$

(it's not super correct, because it uses a specific number of resources, but it should give the general idea). This set is a sub-space of the ambient space, spanned by the ker of $\mathbf{B}$, hence the feasible points in the resource space can be seen as a curve, determined by $\vec{T}$, *extruded* in the $\vec{K}$ direction.

The equi-responsive time sets structure shown in states that a set of points of the resource space is equivalent from the standpoint of responsive time. Since this is the criterion we seek to use, we can define a reasonable and objective equivalence as

Definition 2.0.5. Given $\vec{I}_1$ and $\vec{I}_2$ in $\mathbb{R}^m$,

$$\vec{I}_1 \equiv \vec{I}_2 \quad \Leftrightarrow \quad t_{0\to1} = t_{0\to2} \quad \Leftrightarrow \quad \exists s \in \mathbb{R}^+ : \vec{I}_1 = \vec{I}_2 + s\cdot\vec{K}_{\mathbf{B}}$$

From this we want to obtain a definition of some equivalance class, then a order relationship and finally be able to determine what it means to consume more or less resources. This will prove us that the longer time is achieved when we reach the origin.

A partial order that makes sense, and the set that makes it into a total order.

Theorem 4. If t_f, defined as $t_f \doteq t' \in \mathbb{R}^+$ such that $\int_0^{t'} \vec{\Phi}(t)\mathrm{dt} = \vec{I}_0$ exists, then given any responsive time t_1

$$\int_0^{t_1} \vec{\Phi}(t)\mathrm{dt} = \vec{I}_1 \ll \vec{I}_0 \quad \Leftrightarrow \quad t_1 < t_f$$

Proof. Since we have to meet constraints, we have that

$$\mathbf{B}\int_0^{t_1} \vec{\Phi}(t)\mathrm{dt} = \mathbf{B}\cdot\vec{I}_1 = \int_0^{t_1} \vec{T}(t)\mathrm{dt} \qquad \mathbf{B}\int_0^{t_f} \vec{\Phi}(t)\mathrm{dt} = \mathbf{B}\cdot\vec{I}(0) = \int_0^{t_f} \vec{T}(t)\mathrm{dt}$$

it is intuitive that **if** $\mathbf{B}\cdot\tilde{\mathbf{I}}_1 \ll \mathbf{B}\cdot\tilde{\mathbf{I}}_0$ then

$$\int_0^{t_1} \vec{T}(t)\mathrm{dt} \ll \int_0^{t_f} \vec{T}(t)\mathrm{dt} \quad \Leftrightarrow \quad t_1 < t_f$$

since we assumed that $T(t) \geq 0$ always positive. Then we only need to verify the condition above.
Clearly, if $\mathbf{B}$ was invertible, we would have

$$\mathbf{B} \cdot \vec{I}_1 \ll \mathbf{B} \cdot \vec{I}_0 \quad \Leftrightarrow \quad \mathbf{B}^{-1}\mathbf{B} \cdot \vec{I}_1 \ll \mathbf{B}^{-1}\mathbf{B} \cdot \vec{I}_0 \quad \Leftrightarrow \quad \vec{I}_1 \ll \vec{I}_0$$

Since $\mathbf{B}$ is not invertible, let us verify Eq. above for the generic i-th row. We need to show that

$$B_{i1}I_{11} + \cdots + B_{im}I_{1m} < B_{i1}I_{01} + \cdots + B_{im}I_{0m}.$$

Recall that $I_{1,j} < I_{0,j}$ for each $j \in \{1, \ldots, m\}$. Hence,

$$B_{i1}(I_{11} - I_{0m}) + \cdots + B_{im}(I_{1m} - I_{0m}) < 0,$$

Since $\mathbf{B}_{ij} \geq 0$, for each $i \in \{1, \ldots, n\}$ and each $j \in \{1, \ldots, m\}$.

□

What happens for an unknown $T(t)$?
It is nice to notice that, the equi-responsive time loci depends only on $\int_0^t T(t)\ \mathrm{dt}$, not on the instantaneous requests. Then, we can say that

Lemma 5. Given $\vec{T}_1(t)$ and $\vec{T}_2(t)$ if, for some t_0, t_1,

$$\int_0^{t_0} T_1(t)\mathrm{dt} = \int_0^{t_0} T_2(t)\mathrm{dt} \quad \text{and} \quad \int_0^{t_1} T_1(t)\mathrm{dt} = \int_0^{t_1} T_2(t)\mathrm{dt}$$

Then the *iso responsive time sets* are the same.

This means that if we are able to assume a probability distribution for my $T(t)$, we can use an average as an educated guess, and act as if we knew what the function was.

Formal system design

System design might be seen as the process of making many choices, the effects of which can hardly be assessed independently from one another, in the hope of getting to be best possible system.

Experience and case by case reasoning are essential for the human system engineer. Can we generalize this process so to understand the basic analytic rules underling it and have a computer do it?

A naive approach to the problem would be to find a function $\mathcal{F}$ that, given a system, would tell us how *good* the system is;

$$\mathcal{F} : \text{System design} \rightarrow \text{System evaluation} \tag{3.14}$$

Then, all we are left to do is to try several different systems until we either run out of options or money, and just choose the system with the highest *score*.

This method, although incredibly inefficient, is also incredibly easy to implement, being virtually independent of the structure of the problem at hand. We don't need to know $\mathcal{F}$, we just need to apply it. What is more, not only it doesn't require any assumption on $\mathcal{F}$ (continuity, smoothness etc), but it doesn't even require knowledge of any analytical representation of $\mathcal{F}$. This blind approach is very similar to what biological evolution is based upon. **Unfortunately, usually $\mathcal{F}$ does not exist**.

To see this, let us break down $\mathcal{F}$ in more meaningful *blocks*.

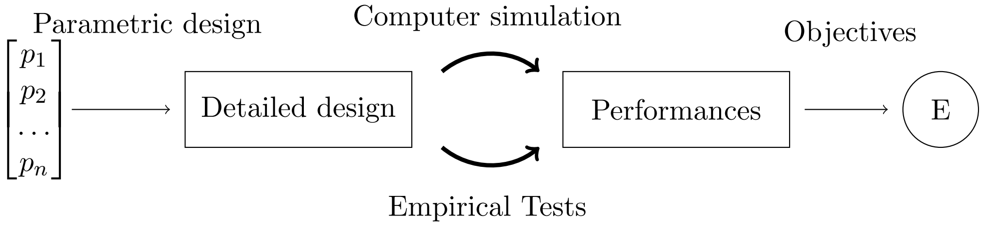


Figure 3.21: A more in depth analysis of $\mathcal{F}$

1. The domain of the function could be a simple list of all the possible design we will try. However, there are good reasons not to do so; first of all, if there is no *continuity* among the various systems, no conclusion can be inferred from one test to the next one, and virtually all possibilities have to be examined.
   A smarter choice will be to use a parametric design; in which a number of parameters defines a design completely.

2. From the detailed design, we can predict or measure performances with either computer simulation (such as CFD or FEM) or empirical tests.

3. Finally, after we have decided which performances we prefer, we can write a fitness function that gives us an evaluation of the Design we have tested.

Example 3.17. A Bolt
We start with the parametric design, which already contains the basic shape of the bolt, with the hexagonal head, the thread, an indication of the material etc. As parameter we may chose, for the geometry body diameter, hexagon diameter, length etc, for the material the concentration of carbon in the steel and so on. Even for this very simple problem, we might want to use numerical simulations or empirical tests if we plan to use it outside the linear domain of the material.
Once we have the performances, we can measure them against our design objectives; if we are interested only in the UTS, rather than on the fragility at low temperature and so on.

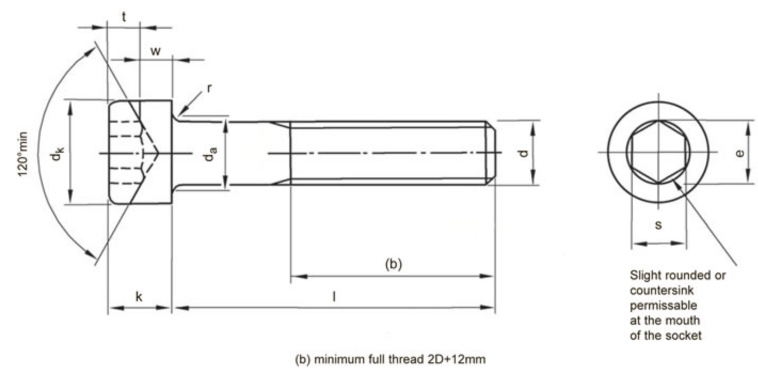


Figure 3.22: A geometrically parametric blot design

The best bolt design will be that which maximizes the fitness function.

Both computer simulation and empirical tests hide the possibility for $\mathcal{F}$ not to be a function, which means that **the same system might be associated with two different evaluations**.

System performances will depend both on system design, system management (how the system behaves), and external conditions. **If we don't specify all three, $\mathcal{F}$ can not be a function**[14], merely a correlation.

While the dependence on external condition might be obvious, that with system management is somewhat new, and has been made possible by modern electronics. As modern systems feature a great deal of software, system behaviour can be reprogrammed and changed at will, with little constraints given by the physical parameters.

Example 3.18. Watt centrifugal Governor
Figure 3.23 shows an analogic control feedback loop. As the Governor spins, centrifugal forces balance with gravitational forces to push outward the two weights. By doing so they open or close the valve that governs the steam engine. There will be a speed above which centrifugal forces win against gravity, therefore closing the valve closes, and vice versa.
To set this speed, one would need to change the gear ratio that connects the governor and the steam engine. This parameter is already included in the physical design, *management* here is not an issue.

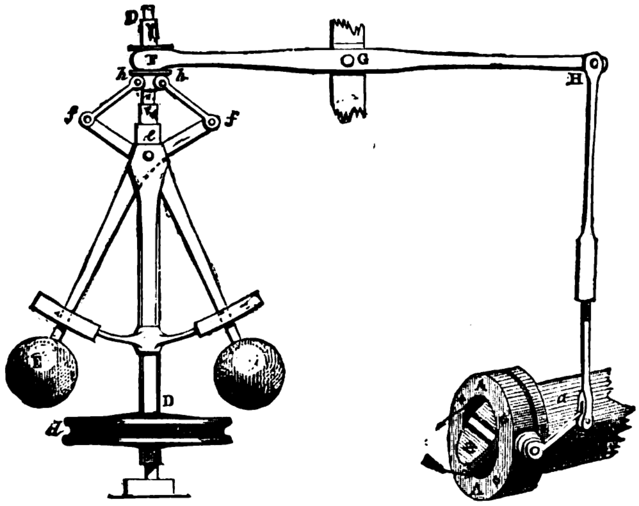


FIG. 4.—*Governor and Throttle-Valve.*

Figure 3.23: Watt centrifugal Governor

Assume we want to find which among two steam locomotive is faster. It is clear that, even if both the engine and external condition are the same, we will not get the same result (hence we won't have a function) if we don't specify the governor geometry.

In the above example however, one might still see the governor's geometry as part of the system design itself, because it takes some non trivial work to change the equilibrium

[14] Since a system design (element of the domain) will not be uniquely linked to a performance level.

speed.
If we examine a more modern example however, the line between system design and management becomes more defined, just as the relationship between system design and system behaviour fades.

Example 3.19. A modern car race

Assume we have two cars and we want to find out which is the fastest around a particular track. Our performance and fitness function are completely defined by the best time around the track. We fix external conditions by having the two car race with the same driver, under the same weather conditions.
This is still not enough, as modern cars feature many settings such as race mode, eco, comfort and so on. We need to specify the settings for each car, for example race mode or eco-friendly mode. If we don't do this, we may find that, under nominally identical conditions, we can obtain different results, which means that $\mathcal{F}$ is not a function.

Let us assume then, that we have specified a management algorithm[15]; we still need to fix external conditions.

If we are dealing with a well understood **component**[16], which has to be used in a bigger system, we will have simple external conditions, in order to give a synthetic representation of the behaviour (UTS for mechanical structure, efficiency of conversion, electrical resistance at a given temperature). This is fundamentally different from what we do when we design a system; the external condition here are hypothesis, which needs to be met during system design. [17]

If we are dealing with complex systems, we want to measure their performances against realistic environmental conditions. This poses the problem of **uncertainty and unpredictable environmental conditions.**

A very reasonable thing to do, in the face of uncertainties, is to handle them statistically. That is to extrapolate the invariant characteristics of the phenomenon, and design to a *reasonable* worst case scenario.

Example 3.20. PSD;
The dynamic loadings on a structure due to wind, acceleration etc can not be accurately predicted. Hence, a smart thing to do is to design to the *highest probable loading*, which is linked to the average loading. This guarantees that the unpredictability of the environmental conditions won't affect system performances.

The uncertainties on environmental conditions might be more significant than those in the case of the dynamic loadings. If we are to enter an age of industrial space, mission oriented design can not be sustainable. Lunch systems already have to satisfy

[15]This has to be developed in the previous chapter; possibly a management algorithm that allows the system to meet all requirements while using the least amount of resources

[16]Systems without management

[17]If we design a structure using a given beam, you know when it will break, and it is our business to make sure the loading does not exceed this condition. When we design a roof, we have to design it to the specification of the ambient it will operate in.

a spectrum of different payload requirement (payload mass, terminal velocity etc) and modular satellite might soon follow. The endeavour of space system design might soon be pursued with a very partial understanding of the system's mission, which increases the scope of the uncertainties considerably.

We could still follow the same approach as before, by designing a system that performs under average performances, or we could use the optimum management to modify our system on a case by case basis.

Example 3.21. A really strange car race

Assume a new race format is instituted, where the teams don't know where the next race will take place until 2 days before the event. The tracks might range from formula one, smooth tarmac to muddy rally terrain, to iced lakes or sandy deserts.
A commercial car might work on all the above surfaces, with little adjustments. This is because it is designed to be good on average on all the surfaces. However, since we have 2 day before each race, we could devise a car that can be rapidly adjusted for the terrain we have to face. *Probably this will be the car with the best overall performances.*

The benefit of designing a reconfigurable system are the following

1. The same system can be used for different missions
2. During operations, the system can make more accurate predictions about real external conditions than what we can speculate during the design phase.

We want to investigate what it means to design a system for maximum reconfigurability over a range of external conditions (rather than maximum average performances over the same range of external condition).

System Design and flexible design concept

We assume that system architecture, as well as overall mission design has been defined in phase A/0. We already know which kind of subsystem we are going to use, their functions and, to some extent, what kind of external conditions and overall system requirement we have to satisfy.
For a standard commercial telecommunication satellite, the architecture is based on the concept of Bus; we know we will need a Power System, an On Board Computer, an ADCS etc. We expect to know something about the orbit in which the system will operate as well as some kind of information on what the system is supposed to do.
Our goal is to analytically derive the **characteristics/requirements at a subsystem level**. Once we have the parameters and characteristics for the standard static system optimization, we will use them as a benchmark for the new, flexible design method.

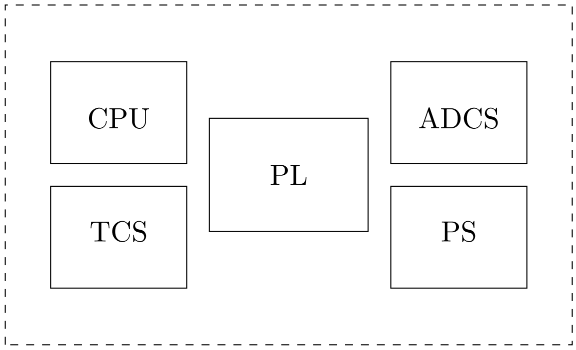


Figure 3.24: A simple satellite architecture.

For each subsystem we need to decide what level of performances (in the broadest sense) we require; what is the conversion efficiency of the PS? What should the capacity of the battery be? How fast can the RW spin? Etc. These are the requirements we need for each subsystem; they can later be used to choose the appropriate components from a catalogue, or in the case of custom subsystems, they would provide the requirement for the preliminary design of each subsystem.

Usually, we think of these parameters as lower boundaries; If the conversion efficiency is higher than the design value, we can expect the system to work better than anticipated. This is not strictly true; even if we ignore the fact that usually increasing one parameters come at a cost for another (in term of mass, economic budget etc), subsystems are usually interdependent or coupled. The requirements for each subsystem are implicit functions of each other. For example, a more efficient conversion in the PS would waste less energy, hence we could use smaller solar panel [18].

This is the reason why it is hard to directly relate system level preferences-objectives

[18] *This is to remind us that the frontier between acceptable system and not acceptable ones is quite complex, non plane curve.*

to subsystem preferences. For example if the imperative is to minimize system mass, we could start by choosing the lightest CPU, then the lightest TCS etc all the way up to the PS. There is clearly no guarantee that the resulting system will be the lightest. On the contrary, one could expect a light system to be quite inefficient, hence requiring a heavier PS[19].

There are many trade-offs to consider, as we have to account for all subsystems interactions. Let us recall the definition of system engineering:

> *A logical process of activities that transforms a set of requirements arising from a specific mission objective in a* ***full description of a system which fulfils the objective*** *in an* ***optimum way****. It ensures that all aspects of a project have been considered and integrated into a consistent whole.*[20]

We rephrase this definition in an equivalent but more formal way: among all system that to are able to meet mission objectives we want to choose the optimal ones, according to *some* criterion. In the space of all the possible systems, we want to find the ones that actually meet mission objectives and, among them the optimal one.
Since a system (at this point, during phase A) is just a collection of subsystems, each with specific parameters, the space of all subsystem can be thought of as a subspace of R^m. We now need to

1. Define the set of possible system.

2. Define the cost function for finding the optimum.

There are three different approaches, either $2 \to 1$, first enforcing the cost function and then checking if the system actually works, $1 \to 2$, defining all the system that might work, and the applying the cost function, finding out which one is the best.
Further more, one might use an hybrid, by solving some mid-point decision by direct optimization. For example, if in a subsystem we have the possibility to either be more efficient or decrease the mass of the subsystem, this choice might be made independently of other subsystems. The first and the hybrid approach might not get to the global optimum, however, they are quite faster. This begin said, the second approach is conceptually better, because it gives more emphasis to the physics of the problem. The optimization is much more volatile; how important is the mass of the system compared with its volume? How much it matter to finish the design phase by the day X versus how much it costs? These are circumstantial considerations and might change during the course of the design phase, due to new technology or other external constraints.

Using the second approach, we want to identify the set of subsystem requirements that make a system *mission-objective worthy.* From architecture and physics, we can

[19] We would have chose the lighter of the viable options, but the options are a function of what we have chosen before!

[20] Stark etc. Space System engineering.

write some[21] balance equations:

$$\begin{cases} \mathbf{N_{CPU}(t)} & = & N_{ADCS}(t)+ & N_{PS}(t)+ & N_{TCS}(t) & + & N_{PL}(t) \\ \mathbf{i_{PS}(t)} & = & i_{ADCS}(t) & + & i_{TCS}(t)+ & i_{CPU}(t)+ & i_{PL}(t) \\ \mathbf{T_{ADCS}(t)} & = & \frac{d^2}{dt}(\theta_{PL}(t)) \cdot I_0 & & + & & T_{ext}(t) \\ \mathbf{q_{TCS}(t)} & = & q_{ADCS}(t)+ & q_{PS}(t)+ & \frac{d}{dt}(T_{PL}(t)) \cdot \overline{mc_p}+ & q_{CPU}(t)+ & q_{ext}(t) \end{cases} \tag{3.15}$$

The first line in Eq 3.15 conveys the necessity for the CPU to perform all the operations requested by the ADCS, by the PS, the TCS and the Payload. Similarly, the PS must be able to supply current to all the subsystem that needs it. For the ADCS and the TCS the effect of external condition also appears and the equations are differential in nature. Note that each function is time dependent and the equations must be verified at all time for the system to be *mission objective worthy*.

If we are able to determine the right hand side of Eq 3.15, **we would have all the parameters for each subsystem that make the system mission objective worthy**.

The maximum authority for each subsystem would simply be the maximum for t taken over the whole mission duration. If we derive 3.15 we can determined how fast the response for each system must be. If we integrate 3.15, we obtain the capacity of the system (how much the battery must hold, maximum Ns for the reaction wheels etc).

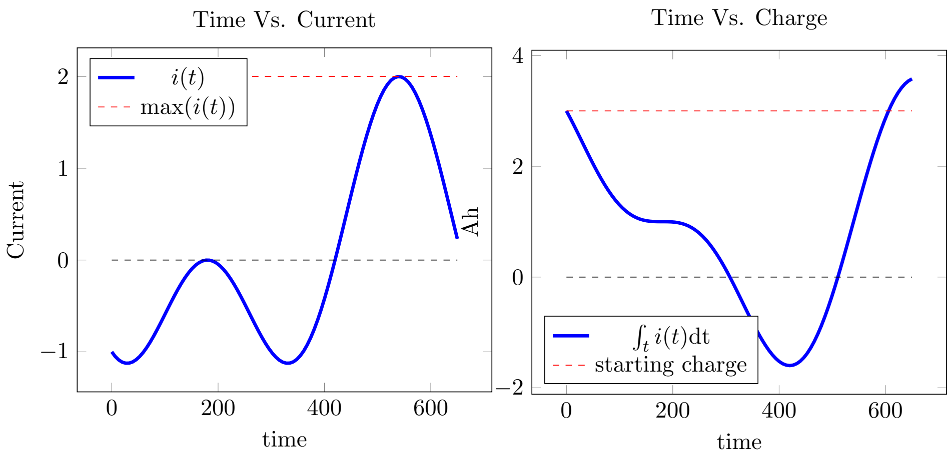


Figure 3.25: Assuming RHS known, we can obtain i_{max} and the battery charge.

However, the right hand side of the balance equation is not known beforehand; it depends on both *external conditions* (such as external heat fluxes of torques and payload requirements) and *internal interdependencies* between subsystems.

While external conditions are given by the mission,(e.g. the amount of sunlight available will be a function only of the orbit we have chosen) internal interdependencies

[21] There can be many, I am just writing the most intuitive ones

are auto-inducing (a bigger ADCS will increase satellite inertia, which will require yet higher level of torque). Taking advantage of this interdependency, we will be able to reduce the variables in the problem.
We begin by dealing with such auto inducing relationships. We can identify two main categories;

1. The effect each subsystem has on other subsystems, which we can call cross-coupling. This is a measure of how much a subsystem relies on other to operate; the TCS uses both computational power and electric power, so it can not operate on its own. Hence a high thermal load will have effects on the PS authority as well as the CPU.

2. The effects that each subsystem has on overall system dynamics (mass/inertial, thermal inertia, ect)

Cross coupling
As a first approximation, let us assume that all subsystem have a substantially mono dimensional output. This is to say that the space of the transformation given by each subsystem is mono dimensional, or that the input output are in a fixed relationship with one another.
Given a level of torque $\mathbf{T_{ADCS}}$, the current i_{ADCS} is uniquely determined; so is computational power requested N_{ADCS}, the waste heat produced q_{ADCS} and so on. This is equivalent to the idea of operational level

$$T_{ADCS}(t) = f_1(x_{ADCS}) \quad i_{ADCS} = f_2(x_{ADCS}) \quad q_{ADCS} = f_3(x_{ADCS}) \quad N_{ADCS} = f_4(x_{ADCS})$$

We assume we are controlling the operational level x_{ADCS} of a subsystem and all outputs/inputs are function of this level. For simplicity, we will write all productions as a function of the most important thing the subsystem produces[22]. Extending this assumption to all subsystem, we can re-write the set of equations 3.15 as

$$\begin{cases} \mathbf{N_{CPU}(t)} & = & N(T_{ADCS})+ & N(i_{PS})+ & N(q_{TCS}) & + & N_{PL}(t) \\ \mathbf{i_{PS}(t)} & = & i(T_{ADCS}) & + & i(q_{TCS})+ & i(N_{CPU})+ & i_{PL}(t) \\ \mathbf{T_{ADCS}(t)} & = & \frac{d^2}{dt}(\theta_{PL}(t)) \cdot I_0 & & + & & T_{ext}(t) \\ \mathbf{q_{TCS}(t)} & = & q(T_{ADCS})+ & q(i_{PS})+ & \frac{d}{dt}(T_{PL}(t)) \cdot \overline{mc_p}+ & q(N_{CPU})+ & q_{ext}(t) \end{cases} \tag{3.16}$$

Now, let us assume that they are also linear; initially we will do this for simplicity, later however we will discuss the range of applicability of this hypothesis. Then, the ADCS can be characterized by its 3 efficiencies

$$N_{ADCS} = \frac{T_{ADCS}}{\varepsilon_{ADCS,N}} \quad i_{ADCS} = \frac{T_{ADCS}}{\varepsilon_{ADCS,i}} \quad q_{ADCS} = \frac{T_{ADCS}}{\varepsilon_{ADCS,q}}$$

We can do this for every subsystem, and obtain the linear system

[22] This is formally allowable because we can expect the designed/most important output, to be a strictly monotone function of the main input, hence invertible.

$$\begin{cases} \mathbf{N_{CPU}(t)} & = & \frac{T_{ADCS}}{\varepsilon_{ADCS,N}}+ & \frac{i_{PS}}{\varepsilon_{PS,N}}+ & \frac{q_{TCS}}{\varepsilon_{TCS,N}}+ & N_{PL}(t) \\ \mathbf{i_{PS}(t)} & = & \frac{T_{ADCS}}{\varepsilon_{ADCS,i}}+ & \frac{N_{CPU}}{\varepsilon_{CPU,i}}+ & \frac{q_{TC}}{\varepsilon_{TC,i}}+ & i_{PL}(t) \\ \mathbf{T_{ADCS}(t)} & = & \frac{d^2}{dt}(\theta_{PL}(t)) \cdot I_0 & & + & & T_{ext}(t) \\ \mathbf{q_{TCS}(t)} & = & q(T_{ADCS})+ & q(i_{PS})+ & q(N_{CPU})+ & \frac{d}{dt}(T_{PL}(t)) \cdot \overline{mc_p}+ & q_{ext}(t) \end{cases}$$

We bring all the term that feature the actions of a subsystem to the left side of the equation.

$$\begin{cases} N_{CPU}(t) & -\frac{i_{PS}}{\varepsilon_{PS,N}} & -\frac{T_{ADCS}}{\varepsilon_{ADCS,N}} & -\frac{q_{TCS}}{\varepsilon_{TCS,N}} & = & N_{PL}(t) \\ -\frac{N_{CPU}}{\varepsilon_{CPU,i}} & i_{PS}(t) & -\frac{T_{ADCS}}{\varepsilon_{ADCS,i}} & -\frac{q_{TC}}{\varepsilon_{TC,i}} & = & i_{PL}(t) \\ 0 & 0 & T_{ADCS}(t) & 0 & = & \frac{d^2}{dt}(\theta_{PL}(t)) \cdot I_0+ \quad T_{ext}(t) \\ -\frac{N_{CPU}}{\varepsilon_{CPU,q}} & -\frac{i_{PS}}{\varepsilon_{PS,q}} & -\frac{T_{ADCS}}{\varepsilon_{ADCS,q}} & q_{TCS}(t) & = & \frac{d}{dt}(T_{PL}(t)) \cdot \overline{mc_p} \quad +q_{ext} \end{cases}$$

The above system can be written more easily in matrix form

$$\begin{bmatrix} 1 & -\frac{1}{\varepsilon_{PS,N}} & -\frac{1}{\varepsilon_{ADCS,N}} & -\frac{1}{\varepsilon_{TCS,N}} \\ -\frac{1}{\varepsilon_{CPU,i}} & 1 & -\frac{1}{\varepsilon_{ADCS,i}} & -\frac{1}{\varepsilon_{TC,i}} \\ 0 & 0 & 1 & 0 \\ -\frac{1}{\varepsilon_{CPU,q}} & -\frac{1}{\varepsilon_{PS,q}} & -\frac{1}{\varepsilon_{ADCS,q}} & 1 \end{bmatrix} \cdot \begin{pmatrix} N_{CPU}(t) \\ i_{PS}(t) \\ T_{ADCS}(t) \\ q_{TCS}(t) \end{pmatrix} = \begin{pmatrix} N_{PL}(t) \\ i_{PL}(t) \\ \frac{d^2}{dt}(\theta_{PL}(t)) \cdot I_0+ \quad T_{ext}(t) \\ \frac{d}{dt}(T_{PL}(t)) \cdot \overline{mc_p} \quad +q_{ext} \end{pmatrix} \tag{3.17}$$

By inverting the above matrix, we can obtain the analytical expression of maximum authority, capacity and response.

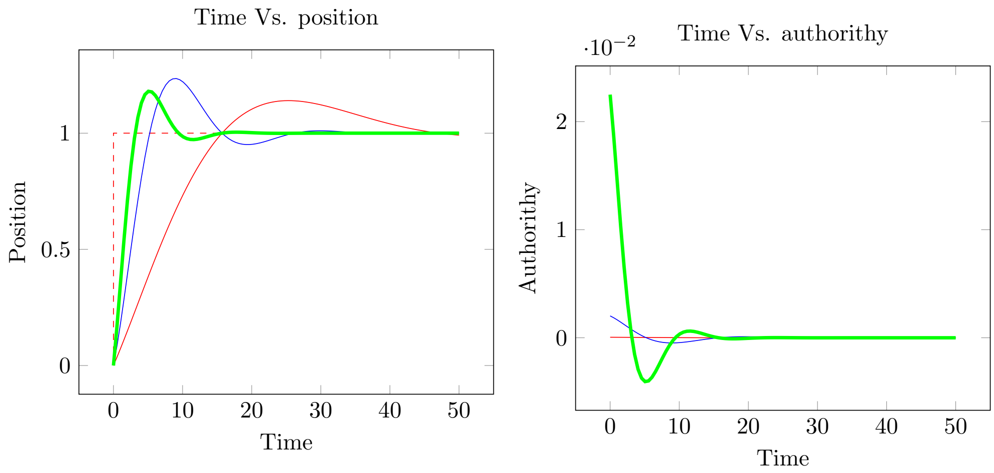
Dynamic effects

We have still some unknown dependent from the choice of subsystems on the right hand side of the equation. The Eq 3.17, satellite inertia I_0, and thermal inertia $\overline{mc_p}$ depend on subsystem parameters. To account for this we could simply iterate; we start with a guess for each subsystem mass, add them up, we obtain the authority for each subsystem, check from a database for new mass and inertia estimates and repeat to convergence. With a reasonable good first guess, the method converges.

Alternatively, if we had a function that was able to link system mass and authority for each subsystem, we could solve the problem directly and analytically.

Optimal control

Some of the equations are differential in nature. If Payload requirement are expressed as function of time (e.g. a tracking manoeuvre $\theta(t)$), one can derive them analytically and there is no real difference between differential equation and algebraic equations. However, it is common for the payload to ask for point to point (P2P) manoeuvres. The PL might require to be pointed in a specific direction in a given time frame, or to be kept within a specific range of temperatures. The path to use to achieve this requirement is not unique, and can be obtain using optimal control theory.



(a) Several path for the same P2P requirement

(b) Correspondent authority required

Figure 3.26: Different solution to optimal control.

Choosing the *right* profile is possible through a series of trade off between how quickly we need to reach the specified value (Fig. 3.26a), how much authority we have to use (Fig. 3.26b), how big is the allowable margin of error is etc. These trade offs are outside the scope of this chapter, and therefore we will always assume these requirements ($\theta(t), T(t)$ etc) to be known.

Imagine that we know **exactly** all external conditions as a function of time. Then we can obtain the exact minimum requirement for our system. We can have the frontier of the *mission objective worthy* system set, as a function of the subsystem efficiencies. In an effort to generalize out notation, we will call $Y_j(t)$ all external conditions and payload requirement for the j-th subsystem and X_j the operational levels of the generic subsystems. Furthermore, we will restrict the problem to a system with only 3 subsystem for simplicity. From Eq 3.17, we have

$$\begin{bmatrix} 1 & -\frac{1}{\varepsilon_{1,2}} & -\frac{1}{\varepsilon_{1,3}} \\ -\frac{1}{\varepsilon_{2,1}} & 1 & -\frac{1}{\varepsilon_{2,3}} \\ -\frac{1}{\varepsilon_{3,1}} & -\frac{1}{\varepsilon_{3,2}} & 1 \end{bmatrix} \cdot \begin{pmatrix} X_1(t) \\ X_2(t) \\ X_3(t) \end{pmatrix} = \begin{pmatrix} Y_1(t) \\ Y_2(t) \\ Y_3(t) \end{pmatrix} \qquad \mathbf{E} \cdot \bar{X} = \bar{Y}$$

which quantifies the link between external request and system characteristics. If we invert[23] matrix $\mathbf{E} \Rightarrow \mathbf{E}^{-1} = \mathbf{B}$, we can analytically derive all subsystem characteristics.

$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix} = \begin{bmatrix} B_{1,1} & B_{1,2} & B_{1,3} \\ B_{2,1} & B_{2,2} & B_{2,3} \\ B_{3,1} & B_{3,2} & B_{3,3} \end{bmatrix} \cdot \begin{pmatrix} Y_1(t) \\ Y_2(t) \\ Y_3(t) \end{pmatrix}$$

[23]The matrix $\mathbf{E}$ is always invertible. This is due to architecture (mostly) and the idea of controllability.

Defining subsystem parameters- Authority of a Subsystem:
As mentioned many times before, we proceed to determine authority for each subsystem as

$$\begin{array}{rcl} \mathrm{Aut}_1 & = & \max_t(B_{1,1}\cdot Y_1(t)+B_{1,2}\cdot Y_2(t)+B_{1,3}\cdot Y_3(t)) \\ \mathrm{Aut}_2 & = & \max_t(B_{2,1}\cdot Y_1(t)+B_{2,2}\cdot Y_2(t)+B_{2,3}\cdot Y_3(t)) \\ \mathrm{Aut}_3 & = & \max_t(B_{3,1}\cdot Y_1(t)+B_{3,2}\cdot Y_2(t)+B_{3,3}\cdot Y_3(t)) \end{array} \tag{3.18}$$

We can now appreciate analytically the effect of the interdependencies between different subsystems. The maximum requirement for each subsystem may be given by a combination of direct and indirect requirements. Furthermore, the timing of the such requests play a vital role. In fact, the same external conditions will not yield the same system design after a time translation. This shows how a statistical approach does not capture the whole phenomenon.

Capacity of a subsystem
The analogous of equation 3.18 for capacity is given by

$$\begin{array}{rcl} \mathrm{Cap}_1 & = & \int_0^{t_1}\{B_{1,1}\cdot Y_1(t)+B_{1,2}\cdot Y_2(t)+B_{1,3}\cdot Y_3(t)\}\,\mathrm{dt} \\ \mathrm{Cap}_2 & = & \int_0^{t_2}\{B_{2,1}\cdot Y_1(t)+B_{2,2}\cdot Y_2(t)+B_{2,3}\cdot Y_3(t)\}\,\mathrm{dt} \\ \mathrm{Cap}_3 & = & \int_0^{t_3}\{B_{3,1}\cdot Y_1(t)+B_{3,2}\cdot Y_2(t)+B_{3,3}\cdot Y_3(t)\}\,\mathrm{dt} \end{array} \tag{3.19}$$

Where the parameters t_i are the minimum time interval in which we need to maintain nominal operations. In the case of the ADCS, this might be the time between desaturation manoeuvres; for the PS, this would be the eclipse time, for a TCS that features PCC, the time in which we would use only passive control.

From these integral, we get informations such as what is the maximum momentum that the RW must handle, what is the capacity of the battery, how much data storage we need, what is the thermal capacity of the PCC etc. Due to the fact that we are taking the integral, a statistical description of the external condition is undistinguishable from a time dependent one.

Dynamic response of a subsystem;
If we derive a time dependent estimation of the external conditions, we can find an upper bound to the system request in term of *reactiveness*. It should be possible to link this parameter to the cut off frequency of the system, but this extends beyond the scope of this chapter.

Cross coupling affects all subsystem requirements, typically requiring higher performances (greater authority, capacity etc). A measure of how much a system is coupled is given by the coefficients of the B matrix.
An ideally decoupled system would have $\mathbf{E} = \mathbf{I} = \mathbf{B}$; the individual request have only a direct effect on the subsystem designed to address them. This might be achieved either with infinite efficiency ($\varepsilon_{i,j} \to \infty$) of by forcefully decoupling the subsystem. We can imagine that a CPU will always need power to operate, however, if there is a power source dedicated to the CPU, they system will be decoupled.

If we extend this idea to all subsystem however, subsystem mass will likely increase a lot, as usually a lot of small subsystems weight more than one big subsystem. This might still be a viable option under the assumption of modular design, in which we sacrifice system performances in order to reduce development costs.

For highly integrated system, we can expect some degree of coupling, which we can optimistically set using the laws of physics (the power needed for a cooling system can be estimated with the reverse Carnot cycle) or information taken from the state of the art.

The case in which external conditions are **known exactly** has the interesting propriety that we can expect the optimum system to be the same under any *reasonable* optimization rules.
We generally can assume that any cost function will be monotone with mass, volume, complexity etc. All these proprieties will also increase monotonously with performances such as authority, efficiency etc. Usually then we can expect that the best system, regardless of the optimum criterion, belongs to the boundary of the mission objective worthy set.
Clearly, this is a viable option only if we know exactly what the external request will be; otherwise we will need some margin of safety, which will place the system within the set (instead of on the surface).

Unknown external conditions
It is to say the least, improbable, that we have exact knowledge of $Y(t)$. The next best thing is therefore to record past missions and use them to extract estimates on external conditions. We use an average value (μ_i) and a dispersion (σ_i). Hence we will model them as a normal distribution $Y_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$. Now we can deal with the **distribution of output levels** for each subsystem;

$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix} = \begin{bmatrix} B_{1,1} & B_{1,2} & B_{1,3} \\ B_{2,1} & B_{2,2} & B_{2,3} \\ B_{3,1} & B_{3,2} & B_{3,3} \end{bmatrix} \cdot \begin{pmatrix} Y_1 \sim \mathcal{N}(\mu_1, \sigma_1^2) \\ Y_2 \sim \mathcal{N}(\mu_2, \sigma_2^2) \\ Y_3 \sim \mathcal{N}(\mu_3, \sigma_3^2) \end{pmatrix}$$

To derive the distributions for X_i, we need to recall some basic results for normal distributions, such as; linear transformation

$$X \sim \mathcal{N}(\mu, \sigma^2) \quad Z = a \cdot X + b \quad a, b \in \mathbb{R} \quad \Rightarrow \quad Z \sim \mathcal{N}(a\mu + b, a^2\sigma^2) \tag{3.20}$$

and sum of independent variables

$$\begin{matrix} X \sim \mathcal{N}(\mu_1, \sigma_1^2) \\ Y \sim \mathcal{N}(\mu_2, \sigma_2^2) \end{matrix} \quad Z = X + Y \quad \Rightarrow \quad Z \sim \mathcal{N}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2) \tag{3.21}$$

Then we have

$$\begin{matrix} X_1 & \sim & \mathcal{N}(& \sum_{i=1}^3 B_{1,i}\mu_i & , & \sum_{i=1}^3 B_{1,i}^2\sigma_i^2 &) \\ X_2 & \sim & \mathcal{N}(& \sum_{i=1}^3 B_{2,i}\mu_i & , & \sum_{2=1}^3 B_{2,i}^2\sigma_i^2 &) \\ X_3 & \sim & \mathcal{N}(& \sum_{i=1}^3 B_{3,i}\mu_i & , & \sum_{i=1}^3 B_{3,i}^2\sigma_i^2 &) \end{matrix}$$

Defining subsystem parameters
For each subsystem, we want to set a maximum output value (**authority**) as well as a **capacity**, or the integral of output with time.

Capacity is straight forward. Due to mission parameters we expect some estimate of the time t_R between refuelling/recharging or general re supply system resources. Hence we have that

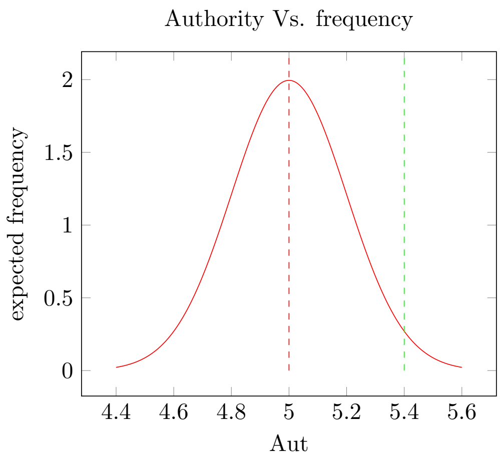
$$C_i \geq \int_0^{t_R} X_i(t)\mathrm{dt} = \int_0^{t_R} \mu_{X_i}\mathrm{dt} = \mu_{X_i} \cdot t_R$$

Provided that t_R is long enough. Using basic results from probability, we can also give an estimate on how accurate this prediction is as a function of the integration time t_r.

Authority for each subsystem
If we applied the method we used in the case of exact knowledge, we would have to estimate the maximum of a normal distribution, which is $+\infty$. Therefore, we must decide on a cover factor, or equivalently, the percent of cases we require the system to be able to answer.
For example, if we decide that it is enough for a subsystem to be responsive in 97% of cases, we can set the maximum authority of the system as

$$\max(X_j) = \mu_{X_i} + 2 \cdot \sigma_{X_i} = \left(\sum_{i=1}^{3} B_{j,i}\mu_i\right) + 2 \cdot \sqrt{\sum_{i=1}^{3} B_{j,i}^2\sigma_i^2}$$

due to other known proprieties of the normal distribution.



Clearly, the more assurance we want, the higher the authority we require. In the same way, the higher the uncertainty we have on the requests (σ_Y) the higher the authority level we require! We can appreciate the effect of the uncertainties in the definition of the maximum authority for the system.

Remark. What is the probability that all three actuator are able to answer to the requests we make? This is also the *percent* of expected time we are able to meet requirements.

It is clearly the probability that all requests are below the maximum authority, which means

$$\mathbb{P}(X_1 \leq \max(X_1) \text{ AND } X_2 \leq \max(X_2) \text{ AND } X_3 \leq \max(X_3)$$

If the event are independent[24] this would be simply the product of individual probabilities

$$\mathbb{P}(X_1 \leq \max(X_1)) \cdot \mathbb{P}(X_2 \leq \max(X_2)) \cdot \mathbb{P}(X_3 \leq \max(X_3))$$

Hard Requirement and House Keeping requirement
Assume that we settle for 97% cover factor; is it enough? Can we allow for 3% of request to go unanswered?

Intuitively, it depends on what kind of requests we are neglecting. If we are saturating some subsystem due to a *incidental* superposition of independent requests, we can easily reschedule them in order to lower instantaneous requirement. For example, if the PS is not able to simultaneously power an attitude adjustment manoeuvre, a demanding temperature control adjustment, data handling etc and all these events are not *functionally linked*[25] we can prioritize some request and queue the rest. We can accept the fact that we are not able to answer such requests because we can avoid them as they happen (furthermore, they are very unlikely[26]).

On the other hand, if the peak is given by a task that **requires simultaneous operations**, failure to comply is not acceptable. For instance, if we require a very accurate tracking, we can not think to power first the ADCS and afterwards the PL; **the two requirements must be addressed simultaneously**.
These request are to be considered known, as they are an key factor in mission design.

To deal with such circumstance, we distinguish between **house keeping request**(HK) which can be handled statistically and **hard requirement**(HR), which are assumed to be known.
For the estimation of systems capacities, nothing changes. The most important results are for authority definition.

Authority with HK and HR
The equivalent of Eq 3.18 uses, instead of the $Y(t)$ the values of each request during the several cases of HR. For each of the n HR conditions obtained from mission design, the system must be able to satisfy

$$\text{Aut} \geq \mathbf{E}^{-1} \cdot Y_{HR,i} \quad \text{for}\, i = 1, ..., n$$

(Where $\geq$ means all components are greater or equal). This requirement can probably be reduced to less than n, as it is clear that if $Y_{HR,i} \ll Y_{HR,j}$ than also $\mathbf{E}^{-1}Y_{HR,i} \ll \mathbf{E}^{-1}Y_{HR,j}$; if we satisfy the jth requirement we have satisfied the ith as well. Furthermore, we assume that all $Y_{HK,i}$ requirement are $\ll$ the $Y_{HR,i}$.

[24] They are not independent, so this is not the case, but it gives the idea
[25] There is no reason why the should be performed at the same time, it just so happens
[26] If they are independent, the probability is the product of individual probability

We then have that, for each subsystem $J = 1, 2, 3$ the required authority will be

$$\text{Aut}_J = \max_i \left(B_{J1}Y_1 + B_{J2}Y_2 + B_{J3}Y_3|_{HR,i} \right) \quad \text{for } i = 1, ..., n \tag{3.22}$$

This process of choosing the maximum value requested individually for usually very different than simply choosing the anti-image of the maximum Y_{HR}. Let us examine two possibilities;

1. By choosing the individual maximum, we find that *all belong to the same event.* That is, we have that we can actually find an Y_{HR} that is bigger than all the other (in the sense of $\ll$). If this is the case, we simply design the system to handle this condition, and everything is analytical and can easily be dealt with.

2. Complementary to the first case, we have that *not all maximum requests come from the same HR*. The maximum authority for each subsystem might be reached only with one HR, while in all other cases, the subsystem uses could use less authority.

The second hypothesis is somewhat more likely than the first, as typically a system is required to perform a multiple objectives maybe very different among each other, which require different things.
This behaviour can be expect to increase as the number of subsystems increases.

Regardless of the case in which we might be, following the above procedure will give us the authority for each subsystem. Using the statistical knowledge on the requests we can assess the needed system capacity. Then we have a pretty good estimate of all the subsystem parameters.
It is important to notice that all the above parameters are function of the specific subsystem efficiencies.

The flexibility approach

The first case does not pose any particular challenge for the design process. The second one however present possibility to improve. In this case, we have that the authority of a subsystem is set by an unusually high demand condition, and in most cases are used to much lower operational levels.

It is worth noticing that this paradigm arises partially from mission requirement, partly from the efficiency of each subsystems. Even with ideal subsystems then, there will always be a mission profile that yield this result (namely, that the system are way over authoritative).

One way to break out of the paradigm above is to relax the hypothesis of mono dimensional production set. We could allow for individual efficiencies to vary, according with instantaneous requests. Clearly, we can not expect to be able to simultaneosly increase **all efficiencies**, otherwise we would have done so in the design phase. A more realistic hypothesis is that we can increase one efficiency at the cost of lowering another. To ask whether this is possible or not is a rather legitimate question. Assume there are

two components that perform the same function (say a power conversion) using different technology, and have been inserted in the system for redundancy. Unless we use the *exact* components, we can imagine that they will exhibit different efficiencies.
Once again, if one subsystem has *strictly higher* efficiencies under every aspect, we might chose that same system twice. The most interesting case is that in which we have two subsystem with non-comparable efficiencies. If this is the case, the overall subsystem efficiency will be the weighted average of the two, (weighted by the amount of output maybe). Hence we can *migrate* from one efficiency condition to the other.
This outcome could be achieved even with the same hardware, but with different algorithms. For example, power control might be obtain with PWM and square waves of more sinusoidal waves.

Whenever this happens, one can modulate the overall subsystem performance by simply using more of one system and less of the other. Although this is very clear if one assumes linear behaviour, for non linear system the effect of this are probably even bigger. **One could even modulate efficiency by using two identical subsystems!**

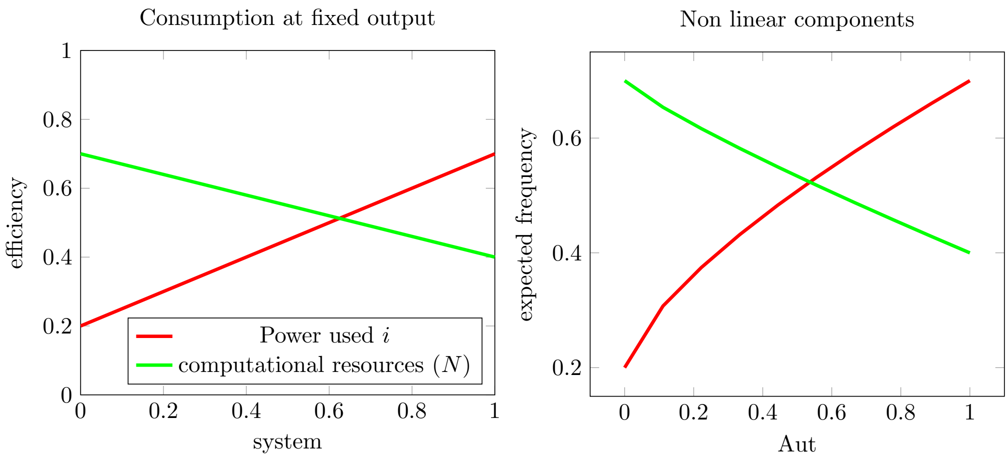


Figure 3.27: Some basic normal distribution to show the cover factor

But what is the benefit of having a subsystem that can change its efficiency?
The intuitive idea is that, if a request is particularly demanding on one subsystem, the other might avoid worsening this condition by increasing the efficiency with regard to the almost saturated subsystem. The downside of this is to increase the load on a different subsystem, but if this is less saturated, it will not develop into a higher authority demand in the design phase.
If this intuition is right, we can lower over all authority on all subsystems, independently of the technology used, and without sacrificing any system properties[27].

Before studying this method analytically, we wanted to obtain some semi-empirical results. We are interested in the decrease in authority between the static design and the

[27] The flexible system is able to do everything the static system can do, but with smaller actuators

dynamic one.
The static design works like this:

1. Fix efficiencies for all subsystems (6 variables)
2. Fix External conditions, such as maximum requests and usual requests (6 variables)
3. Obtain Authority for each subsystem (3 values) using 3.22

For the dynamic case the process is

1. Fix efficiencies for all subsystems (6 variables)
2. Fix flexibility, how much I can improve or decrease each efficiency (6 variables)
3. Fix External conditions, such as maximum requests and usual requests (6 variables)
4. Obtain Authority for each subsystem (3 values) by brute force.

We then compare the authority of systems with the same efficiency and external conditions.

How efficiency changes
We allow for efficiencies to change, then the matrix $\mathbf{E}$ will be something like

$$\mathbf{E}_{var} = \begin{bmatrix} 1 & -\frac{1}{\varepsilon_{1,2}\pm\delta\varepsilon_{1,2}} & -\frac{1}{\varepsilon_{1,3}\pm\delta\varepsilon_{1,3}} \\ -\frac{1}{\varepsilon_{2,1}\pm\delta\varepsilon_{2,1}} & 1 & -\frac{1}{\varepsilon_{2,3}\mp\delta\varepsilon_{2,3}} \\ -\frac{1}{\varepsilon_{3,1}\mp\delta\varepsilon_{3,1}} & -\frac{1}{\varepsilon_{3,2}\mp\delta\varepsilon_{3,2}} & 1 \end{bmatrix}$$

Where the fact that we can not increase **all efficiencies** for a given subsystem is given by the inverted sign of the $\pm$. To quantify these change in efficiencies, we use the variables p, q, r all in $[-1, 1]$. The matrix $\mathbf{E}_{var}$ is now

$$\mathbf{E}_{var}(p,q,r) = \begin{bmatrix} 1 & -\frac{1}{\varepsilon_{1,2}+q\cdot\Delta\varepsilon_{1,2}} & -\frac{1}{\varepsilon_{1,3}+r\cdot\Delta\varepsilon_{1,3}} \\ -\frac{1}{\varepsilon_{2,1}+p\cdot\Delta\varepsilon_{2,1}} & 1 & -\frac{1}{\varepsilon_{2,3}-r\cdot\Delta\varepsilon_{2,3}} \\ -\frac{1}{\varepsilon_{3,1}-p\cdot\Delta\varepsilon_{3,1}} & -\frac{1}{\varepsilon_{3,2}-q\cdot\Delta\varepsilon_{3,2}} & 1 \end{bmatrix}$$

And $\Delta\varepsilon_{i,j}$ is now the maximum efficiency change the subsystem can perform. While the parameters p, q, r will have to be set by the internal management of the system, as the mission progresses, in real time, the parameters $\Delta\varepsilon_{i,j}$ are set during the design phase.

It is worth noticing that using this model, since by choosing $p, q, r = 0$ we revert to the static case, we can not make things worse. The question is whether or not we are able to **improve more** than the cost (in term of mass) of having a flexible system.

There are two more things to consider; Maybe we can not have higher efficiencies for each subsystem because we have reached the state of the art. If we haven't though, a static optimization yields no results, because on average, it doesn't matter if you increase one efficiency to lower the other.

How do we chose which authority is the best one?
This is a problem just for this specific exercise, and in the real application can be overlooked. Once a target function is defined, there is no longer ambiguity in the definition of *best authority*. We will have to invert $\mathbf{E}_{var}$, so for easier notation we rename everything.

$$\mathbf{E}_{var}(p,q,r) = \begin{bmatrix} 1 & -a_{1,2}(q) & -a_{1,3}(r) \\ -a_{2,1}(p) & 1 & -a_{2,3}(r) \\ -a_{3,1}(p) & -a_{3,2}(q) & 1 \end{bmatrix} \tag{3.23}$$

We need to clarify how we chose the authority in the dynamic case(step 4). A simple, brute force way to do it is as follows; The anti image of the ith HR is given by

$$\text{Aut}_{:,i} = \mathbf{E}_{var}(p,q,r)^{-1} Y_i$$

We want to lower all subsystem authority[28]. We can cycle through a lot of possible combinations of p, q, r and find the one we like the best. In this case, we chose according to the $\ll$ preference, hence for each HR, we have a vector containing the authority for each actuator $\text{Aut}_{HR,j}$. Due to how we chose it, we are sure that there is no other vector (and p, q, r combination) that is able to match $Y_{HR,j}$ and is strictly smaller than our choice.
Once we have this for all the n HR, we set individual authority as before

$$\text{Aut}_J = \max_i (\text{Aut}_{J,i}) \quad \text{for}\, i = 1, ..., n$$

What kind of improvement can we expect?

This model has a lot of variables. To guide our analytical search, we performed a Monte Carlo analysis. We have all 18 variables change within some reasonable intervals and record the results.

Simulation Parameters:
Subsystems range of characteristics

[28]This is very demanding, but doesn't require a trade off.

Subsystem Characteristics		Efficiency Min	Efficiency Max	Flexibility Min [%]of eff	Flexibility Max [%]of eff
PS	Heat	83%	95%	1	50
	Computational [MHz/W]	0.1	5	1	50
TCS	Power	50%	90%	1	50
	Computational [MHz/W]	0.1	5	1	50
CPU	Power [W/MHz]	0.01	0.1	1	50
	Heat [W/MHz]	0.01	0.1	1	50
External Conditions		Mean Min	Mean Max	Peak Min	Peak Max
Current	[W]	5	50	2	10
Waste heat	[W]	5	50	2	10
Payload data handling	[MHz]	30	300	2	10

Figure 3.28: All the combinations tried in the Monte Carlo method.

The result of running the program on the above simulation is shown in figure 3.29. The x axis represents the percent decrease in the authority requirement for each subsystem. This can be also understood as an additional safety factor that we would have if we designed the system with the classical method vs the proposed one. The y axis shows how frequent any particular value has been obtained.

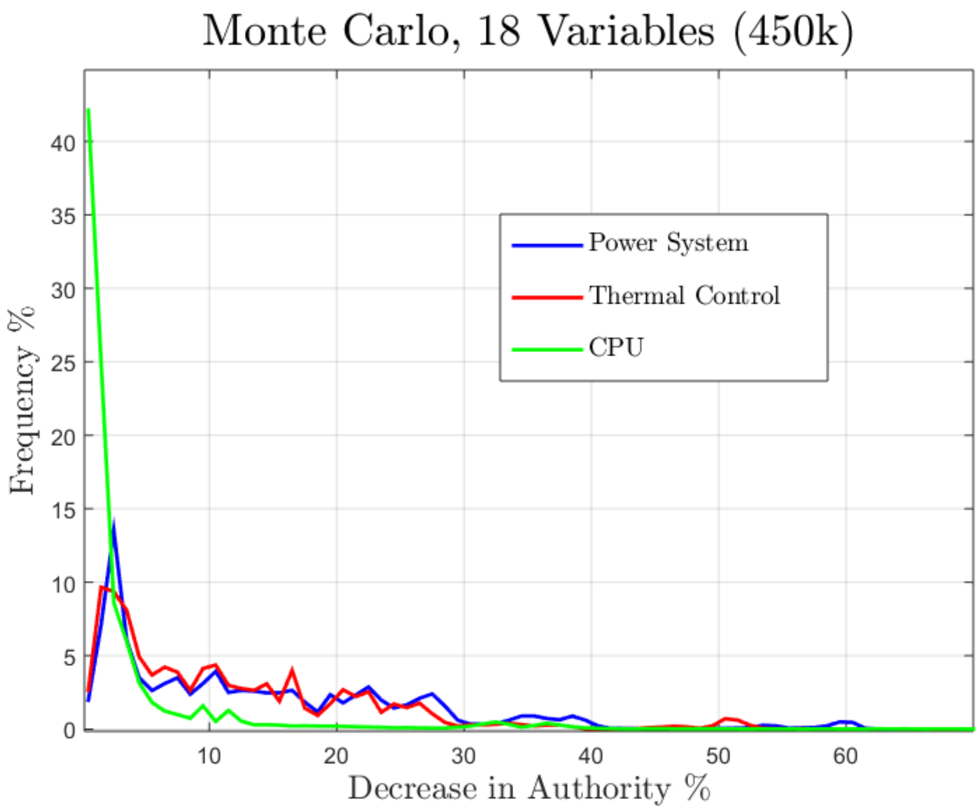


Figure 3.29: Changing all 18 variables.

Figure 3.29 shows a rather jagged, although consistent pattern. However, given the number of variables, it is very hard to discern what is meaningful and what is not. In an effort to give more meaningful results, we fixed the system characteristics and cycled through many different mission requirement.

System A; subsystems characteristics

- PS: Energy conversion efficiency between 82% and 87% (heat output). **Static 84.5%**
  Computational cost per W between 0.08 MHz and 0.12 Mhz. **Static 0.10**

- TCS: power required for 1 W of heat removed 0.8W and 1.2W. Eff 44% and 55% (Combination of Peltier cells and other methods).**Static 50%**
  Computational cost per W removed 0.08 MHz and 0.12 Mhz.**Static 0.10**

- OBDH: power required for 1Mhz 0.008 W and 0.012 W. **Static 0.010**
  Heat removal per Mhz 0.08 W and 0.12 W. **Static 0.010**

External Conditions		Mean		Peak	
		Min	Max	Min	Max
Current	[W]	10	100	2	6
Waste heat	[W]	10	100	2	6
Payload data handling	[MHz]	20	80	2	6

Figure 3.30: External condition used in figure 3.31

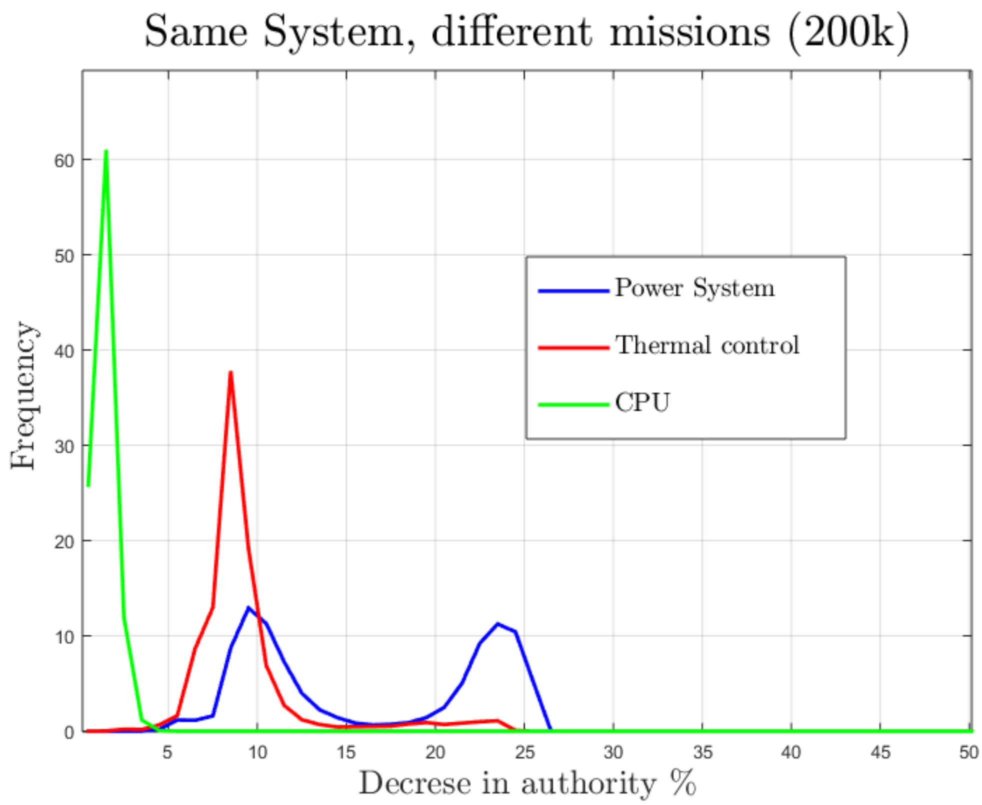


Figure 3.31: Using system A, we vary mission parameters to assess % gains.

The most interesting peaks might seem a relatively small. However, 10% of 200 000 missions is still 20 000 mission in which we could decrease up to 25% the authority of the PS without and *functional* drawback! Again, this improvement is almost regardless of the technology we are using, so it is a big improvement on every subsystem.

The data we have used for the simulations above were taken by data sheet from cubesat shop. Their relevance however, remains questionable. Not everything is available and there is no information on how system proprieties would scale etc.
These simulation however were not performed with the intent of getting accurate or reliable data, just indications for the analytical model. The main question however is,

how much flexibility is needed, and is it achievable?
This is the main point because without flexibility, the two approached are exactly the same.

Flexibility of a linear subsystem

Let us set aside the schematic, functional representation shown in figure 3.24 and return to the component based one (3.32). While the first one was useful to show the functional relationships between the subsystems, this has the advantage of showing the actual, physical subsystems, with their redundancies and interconnections. We will use it to show more intuitively how flexibility can be obtain from redundancy.

Assume we have a system $\mathcal{S}$ with n subsystems, L goods (with $L < n$) and a well defined management algorithm $X(\vec{R})$.

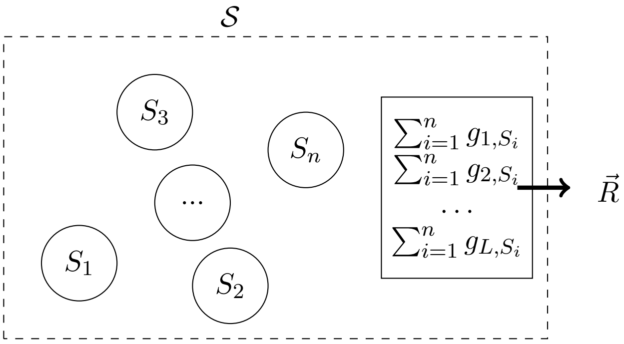


Figure 3.32: A generic system

We describe such system with an adjoint production function $\vec{S}(X) : [0,1]^n \to \mathbb{R}^L$ to model production and a adjoint consumption function $\vec{Q}(X) : [0,1]^n \to \mathbb{R}^L$. The first provides the outputted amount of goods for every possible operational level X while the second quantifies the amount of resources that have been consumed from internal resources. Note that the second is somewhat unnecessary, as all the information about consumption are contained in $\vec{S}$. We define it only to emphasize consumption.

Assume also that each subsystem production set can be written as a linear function of its operational level,

$$S_i = \begin{pmatrix} g_{1,i}(x_i) \\ g_{2,i}(x_i) \\ \cdots \\ g_{L,i}(x_i) \end{pmatrix} = \begin{pmatrix} \alpha_{1,i} \\ \alpha_{2,i} \\ \cdots \\ \alpha_{L,i} \end{pmatrix} \cdot x_i \quad x_i \in [0,1],\ \alpha \in \mathbb{R};$$

Then,we can write

$$\vec{S}(X) = \begin{pmatrix} g_{1,S_1}(x_1) + g_{1,S_2}(x_2) + \cdots + g_{1,S_n}(x_n) \\ g_{2,S_1}(x_1) + g_{2,S_2}(x_2) + \cdots + g_{2,S_n}(x_n) \\ \cdots \\ g_{L,S_1}(x_1) + g_{L,S_2}(x_2) + \cdots + g_{L,S_n}(x_n) \end{pmatrix} = \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & \dots & \alpha_{1,n} \\ \alpha_{2,1} & \alpha_{2,2} & \dots & \alpha_{2,n} \\ \cdots & \cdots & \cdots & \cdots \\ \alpha_{L,1} & \alpha_{L,2} & \dots & \alpha_{L,n} \end{bmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{pmatrix} = \vec{R}$$

$$\vec{S}(X) = B \cdot X \qquad B \in \mathbb{R}^{L,n}$$

Now, the set of all the possible X that satisfy the management problem is very easy to identify

$$\vec{S}(\vec{R})^{-1} = \left\{ X = X_p + X_k, \text{ where } B \cdot X_p = \vec{R} \text{ and } X_k \in \ker(B) \right\} \text{ for } X, X_p, X_k \in [0,1]^n$$

$\vec{S}^{-1}$ can be easily represented on a vector space as an hyperplane with dimension $n - L$.

Example 3.22. Assume a system composed of 2 subsystem ($n = 2$) that co-operate to produce only one good ($L = 1$). We have $\dim(\mathrm{Ker}(B)) = 1$ and $\vec{S}^{-1}$ will be a segment on the plane.

$$B = [\alpha_1 \quad \alpha_2]; \quad B \cdot X = R \quad \Rightarrow \quad X_p = \begin{pmatrix} 0 \\ \frac{R}{\alpha_2} \end{pmatrix} \quad X_k = \begin{pmatrix} 1 \\ -\frac{1}{\alpha_2} \end{pmatrix}$$

Or, in the analytical description

$$x_2 = \frac{R}{\alpha_2} - \frac{\alpha_1}{\alpha_2} x_1$$

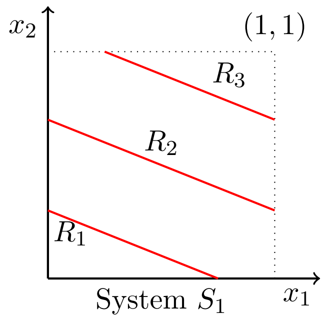


Figure 3.33: Representation of $\vec{S}^{-1}(R)$

Independently of the required $\vec{R}$, the slope, or direction of $\vec{S}^{-1}$ is determined by $\ker(B)$.
Notably, this is a characteristic of the system, not of any specific subsystem; it tells something about the proprieties of the ensemble.

On the same space $\mathbb{R}^n$, we have defined the cumulative consumption function $\vec{Q}$. Here we have several goods that might be consumed for each point X in the operational space. **As we have seen in chapter 1 and 2**, setting an equivalence between these goods is a demanding task. Therefore, we refrain from making such a demanding decision, and plot each consumption function

$$\vec{Q}(X) = \begin{pmatrix} g_0(x_1) + g_0(x_2) + \dots \\ \dots \\ g_L(x_1) + g_L(x_2) + \dots \end{pmatrix} = \begin{pmatrix} G_0(X) \\ \dots \\ G_L(X) \end{pmatrix}$$

Without having to say anything about the type of management we intend to use, we can say that it will depend on these quantities G_i. Let $G_0(X) : \mathbb{R}^n \to \mathbb{R}$ be the consumption of a given internal finite good from all the subsystems.

We initially assume that $G_0(X)$ is linear as well; To avoid a trivial system (with mere reservoirs) we assume that *it looks something like this*

$$\vec{S}(X) = \begin{bmatrix} \alpha_1 & \alpha_2 & 0 & 0 \\ -\beta_{1,1} & -\beta_{1,2} & \beta_{1,3} & 0 \\ -\beta_{2,1} & -\beta_{2,2} & 0 & \beta_{2,4} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} R \\ 0 \\ 0 \end{pmatrix} \qquad \alpha_{i,j} > 0$$

The system above has only one direct output (R) which can be produced using either S_1 or S_2 (the only systems that can produce output on the first line). These however consume two resources (which we'll call G_1 and G_2) and are supplied by the S_3 and S_4. Once we determine x_1 and x_2, it is quite obvious that we don't have any choice for x_3 and x_4, hence the figure above are still a valid representation $R(x_1, x_2)$. We can write G_1 and G_2 as function of x_1 and x_2.

$$\begin{cases} G_1(X) = \beta_{1,3} \cdot x_3 \\ \beta_{1,3} \cdot x_3 - \beta_{1,1}x_1 - \beta_{1,2}x_2 = 0 \end{cases} \quad \to \quad G_1(X) = \beta_{1,1}x_1 + \beta_{1,2}x_2 \qquad x_2 = \frac{G_1}{\beta_{1,2}} - \frac{\beta_{1,1}}{\beta_{1,2}}x_1$$

And in the same way $G_2(X) = \beta_{2,1}x_1 + \beta_{2,2}x_2$. Let us plot $R(X)$, $G_1(X)$ and $G_2(X)$ on the same domain (now we assume the domain to be (x_1, x_2)).

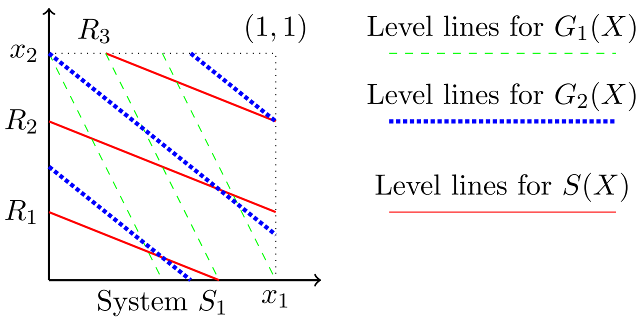


Figure 3.34: Level lines for both $\vec{S}$, G_1 and G_2 in a simple case.

In figure 3.34 we see that both subsystems S_1 and S_2 can use some amount of the resource G_1, G_2 to produce the desired output. The management algorithm will chose an $X^\star \in \vec{S}^{-1}(R)$ to optimize $\vec{Q}(X^\star)$, hence the trade off between consumption of G_1 and G_2 in some yet unspecified sense. In general, we can not, and do not want to, say anything more on the preferred value of $\vec{Q}(X)$.

Intuitively, if we have at our disposal 2 resources G_1, G_2 that can both be converted into a needed output, using more of one will reduce the consumption of the other. Hence, if one is more valuable than the other, we will want to maximize consumption of the less valuable one.

The management will have to chose a point in $\vec{S}^{-1}(R)$. We want to emphasize the possibility we have. Since in this case the set $\vec{S}^{-1}(R)$ is a line we use a parametrization

$$X(t) \text{ such that } \forall X \in \vec{S}^{-1}(R')\, \exists\, t \in [0,1] : X(t) = X$$

Which in the linear case, can be written as

$$\vec{S}^{-1}(R) = \begin{pmatrix} x_1(t) & = & \gamma t + K \\ x_2(t) & = & \frac{R}{\alpha_{1,2}} - \frac{\alpha_{1,1}}{\alpha_{1,2}}(\gamma t + K) \end{pmatrix} \gamma, K \text{ such that } x_1(t=0) = ... \quad x_1(t=1)$$

Where K, γ depend only the level of R requested. Now we can plot $G_1(X), G_2(X), X \in \vec{S}^{-1}(R)$.

$$G_1(X(t)) = \beta_{1,1}x_1(t) + \beta_{1,2}x_2(t) = \beta_{1,1}(\gamma t + K) + \beta_{1,2}\left[\frac{R}{\alpha_2} - \frac{\alpha_1}{\alpha_2}(\gamma t + K)\right]$$

$$G_1(t) = \left(\beta_{1,1} - \beta_{1,2}\frac{\alpha_1}{\alpha_2}\right)\gamma \cdot t + \left(\beta_{1,1}K + \beta_{1,2}\frac{R}{\alpha_2} + \beta_{1,2} \cdot \frac{\alpha_1}{\alpha_2}K\right)$$

We obtain the same results with $G_2(t)$, we just need to substitute $\beta_{1,1}$ and $\beta_{1,2}$ with $\beta_{2,1}$ and $\beta_{2,2}$.

$$G_2(t) = \left(\beta_{2,1} - \beta_{2,2}\frac{\alpha_1}{\alpha_2}\right)\gamma \cdot t + \left(\beta_{2,1}K + \beta_{2,2}\frac{R}{\alpha_2} + \beta_{2,2} \cdot \frac{\alpha_1}{\alpha_2}K\right)$$

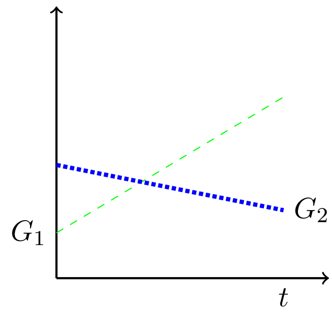


Figure 3.35: Consumption of G_0 and G_1 on an iso R line

Since the management algorithm is merely choosing a t, we are interested in knowing weather resource consumption increase or decreases with t. Since these are linear functions, this is straight forward.

$$\frac{dG_1}{dt} = \left(\beta_{1,1} - \beta_{1,2}\frac{\alpha_1}{\alpha_2}\right)\gamma \qquad \frac{dG_2}{dt} = \left(\beta_{2,1} - \beta_{2,2}\frac{\alpha_1}{\alpha_2}\right)\gamma$$

We divide our options in two possibilities, either G_1' and G_2' have the same sign, or they don't.

1. If they have the same sign (they both either increase or decrease with t), there is no trade off;Assume for simplicity that $G'_1, G'_2 > 0$

$$\beta_{1,1} > \beta_{1,2}\frac{\alpha_1}{\alpha_2} \qquad \beta_{2,1} > \beta_{2,2}\frac{\alpha_1}{\alpha_2} \quad \Rightarrow \quad \frac{\alpha_2}{\beta_{1,2}} > \frac{\alpha_1}{\beta_{1,2}} \qquad \frac{\alpha_2}{\beta_{2,2}} > \frac{\alpha_1}{\beta_{2,2}}$$

   Notice that the ratios $\alpha_j/\beta i, j$ are the efficiencies with which the subsystem S_j converts the resource i into the required good. In this case, the subsystem 2 is **strictly more efficient** than 1. When asked to produce R', it will consume less G_1 and less G_2 that what S_1 would require.
   This is not a very interesting example, in the design phase we would simply take out the less performing subsystem.

2. In light of what shown above, we will assume that $G'_1 \cdot G'_2 < 0$. This means that to produce the same output, one subsystem consumes more G_1 than the other (and vice versa for G_2).

Flexibility of a non linear subsystem

Two identical linear subsystem are not reconfigurable/flexible; this is obvious from the definition of linearity. Given the output, the input is fixed

$$f_1(x_1) + f_1(x_2) = O_1 \to c_1 x_1 + c_1 x_2 = O_1 \to x_1 + x_2 = \frac{O_1}{c_1}$$

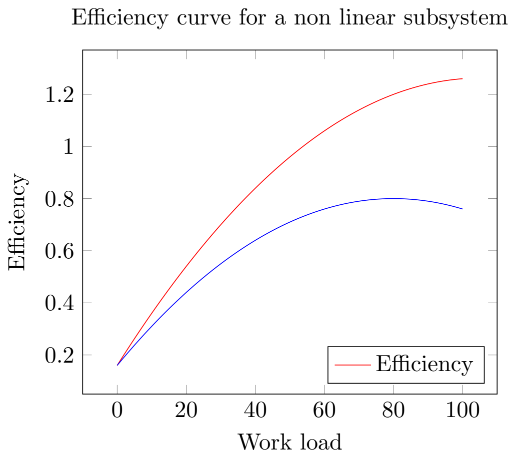
As soon as we lift the hypothesis of linearity however, we can use two identical subsystems and maintain some flexibility. The simplest production function we can think of is a polynomial

$$f_1(x_1) = c_2 \cdot x_1^2 + c_1 \cdot x_1 + c_0$$

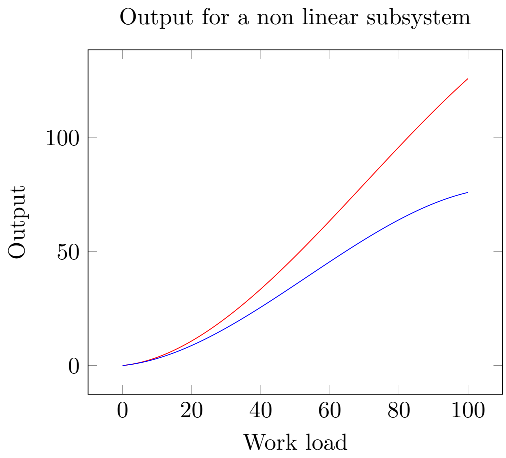
Now, even if the system has two identical subsystems, we the anti image of the wanted output is not unique.

$$f_1(x_1) + f_1(x_2) = O_1 \to c_2 \cdot (x_1^2 + x_2^2) + c_1 \cdot (x_1 + x_2) + c_0 = O_1$$

Which can be used to implicitly define a locus in which the above is verified, or the options we have at our disposal and from which we can choose.



(a) Efficiency



(b) Production function of each subsystem

Effect of Changes in Dynamic Parameters

Now we want to offer some ideas on how to account for changes that affect the dynamic characteristics as well as the functional one. A classical example would be a trade off between subsystem mass and efficiency; is the increase in efficiency worth the increase in mass?

To be more specific, by dynamic characteristics we mean every constant that influences the differential equations 3.15, such as mass, angular inertia, thermal conductivity, electrical charge capacity etc etc. Adding a new subsystem or modifying an existing one will usually change many of these proprieties; since our adjustment are aimed at maximizing management effectiveness, the minimum, *mildest* condition we put on dynamic changes is *not to make the system worse.* We can interpret this desire to not worsen our condition as follow

The system after the change should be able to do everything the system was able to do before the change.

To formally state the above, let us begin by focusing on a mono dimensional dynamic system. The generic differential equation will depend on a series of dynamic parameters $M = (m_1, m_2, \dots, m_n) \in \mathbb{R}^n$, where n is the order of the differential equation. To be as generic as possible[29] we will identify it as

$$f : C^1 \times \mathbb{R}^n \to \mathbb{R} \qquad f(x(t), M) = u(t)$$

Usually, the problem of dynamics is aimed at solving for $x(t)$ given the external input $u(t)$. In this instance however, we are only concerned with what we can make our system do. The question is, given $x(t)$, can we produce a $u(t)$ able to guide our system on $x(t)$?

[29]We are assuming all physical quantities to be continuous

If we assume that our actuators don't suffer any dynamic effects (zero order[30]) we only need to avoid saturation of the actuators. That is, we need to be sure that all the $u(t)$ that we need to keep the system on the wanted trajectory $x(t)$ are below the threshold of maximum output, which is a characteristics of the system.

Example 3.23. Imagine a simplified model of a satellite as a rigid body on a plane, in a low orbit environment. We wish to rotate it of an angle θ_t, according with the law $\theta(t), t \in [t_0, t_1]$. The coefficients $\vec{M} = (I, c)$ are its inertia along the axis of rotation and a drag coefficient to account for some atmospheric residue. The differential equation that describes its dynamic could be something like

$$I\ddot{\theta} + c\dot{\theta} = u(t)$$

Where $u(t)$ is the cumulative torque which the system is able to produce. This might be the sum of the torque given by the reaction wheels, magneto torquers and thrusters. Again this is a characteristic of the system as a whole.
Given $\theta(t)$, the manoeuvre will be feasible if

$$\max\left\{I\ddot{\theta}(t) + c\dot{\theta}(t)\right\} \leq U \quad \forall t \in [t_0, t_1]$$

where U is the maximum achievable output $u(t)$, or the actuator authority.

Note that, given $x(t)$, finding the maximum $u(t)$ required to accomplish the manoeuvre is trivial. In light of this, the extension to non linear dynamics (for once) is also trivial. Since we are going to use the maximum quite often, we define a semi-norm as

Definition 3.0.6. Let $f(t) : \mathbb{R} \to \mathbb{R}$ then

$$||f(t)|| = \max\{f(t)\}$$

Remark. It is a norm if $f(t) \to [0, +\infty)$, it is a semi norm if $f(t) \to \mathbb{R}$ since if $||f(t)|| = 0$ does not imply $f(t) = 0$ for any t.

Then, we can formalize our requirement;

Definition 3.0.7 (Possibility set)**.** Let $\vec{M} = (m_1, m_2, \ldots, m_n) \in \mathbb{R}^n$ be the vector of dynamic coefficients, $U \in \mathbb{R}$ the value of the maximum output feasible with our system and $f(M, x)$ be the *differential equation.* We define as the **possibility set**

$$\mathbb{P}_{\vec{M},U} \doteq \left\{x(t) \in C^1(\mathbb{R}) \text{ such that } ||f(x, \vec{M})|| \leq U\right\} \tag{3.24}$$

If we assume that the system dynamics f doesn't change, it is clear that

$$U_1 \leq U_2 \quad \Rightarrow \quad \mathbb{P}_{\vec{M},U_1} \subseteq \mathbb{P}_{\vec{M},U_1}$$

We are interested in the inclusion relation as we change $\vec{M}$. The intuitive notation we had before can be expressed formally as;

[30] We can produce any output, without having to worry about how fast it's rising or falling and so on

Definition 3.0.8 (Allowable changes)**.** Given a system with dynamic vector $\vec{M}_1$ and maximum output U_1, **a change in the system is allowable** if the possibility set of the resulting system $\vec{M}_2, U_2$ include the previous.

$$\mathbb{P}_{\vec{M}_1,U_1} \subseteq \mathbb{P}_{\vec{M}_2,U_2} \tag{3.25}$$

Remark. Note that:

1. As mention before, checking if any one $x(t)$ belongs to any $\mathbb{P}_{\vec{M},U}$ is remarkably simple, even for complex, non linear system. However if we want to define a possibility set by testing every possible $x(t)$, we need to devise a more strategic test.

2. It is clear that most $\mathbb{P}_{M,U}$ are infinite, with the notable exception of the free motion possibility set $\mathbb{P}_{M,0}$. Therefore, proving the inclusion by trying every possible combination is in general not feasible.

3. It is also true that we might not want/need to maintain, in the new design, all the trajectories $x(t)$ that were available in the previous system. However, since we have not found any reasonable way to determine which trajectories are superfluous, we will assume that none are.

An engineering approach to the problem might be to give an estimate of a number of the most demanding manoeuvre and then checking for those. It is not formally complete, but it is very simple.
Otherwise, for a linear system, a more sophisticated strategy may be the following;

1. Under *reasonable assumptions* for x, we can decompose it in Fourier series.

2. We can analytically find the $||f(M,x)|| = u$ for a generic harmonic.

3. We add them all up. We can say something like: if the PSD of x is such an such, then $||f(M,x)|| < U'$. This way we can characterize the whole possibility set with only one function, PSD.

4. Using the same PSD we can see how U' changes as M changes.

5. We confront the two proposed systems.

Parametric Design

As more of a curiosity than an actual suggestion, we want to consider why parametric design might be extremely interesting for the flexible design concept. We will return to the most basic idea of design method.

If our subsystem choice is bounded to a finite set, we could actually try every possible system and chose the (*loosely defined*) best one. Even with tomorrow's computational power however, this method is bound to saturate our computational capabilities extremely quickly.

The possible design to test for a given problem are given by

$$\prod_{i \in \mathcal{S}} \# \left(\mathcal{S}_i \right),$$

where $\mathcal{S}$ is the set of the subsystems (which we assume to be finite) and $\mathcal{S}_i$ is the set of the possible choices for the subsystem i. If one think about a simple system, with 5 subsystem and 5 possible choices, this yields 5^5 possibilities. Considering the computational burden of every simulation, which must find the optimal management within a range of different requirements and external conditions, this is only theoretically a viable option.

Furthermore, there are good reasons to use this method with parametrized components, instead of modular ones. Hence we would have a continuous range of different subsystems. In this second case, even the theoretical possibility vanishes.

From the finite subsystem space to continuous; parametric design
Parametric design is the idea that we understand a subsystem well enough to be able to write a set of functions that fully describes its characteristics, and that can be met during manufacturing.

Example 3.24. Imagine a hypergolic bipropellent propulsion system for space application. Its efficiency can be measured by the characteristic velocity

$$c = \frac{Thrust}{\dot{m}_{fuel}} = c^{\star} \cdot c_F$$

Where c_F depends on the expansion ratio of the divergent nozzle, the $c^{\star}$ can be linked with the length of the chamber, as longer chambers allow for a more complete combustion.
With appropriate testing, it is possible to write c a function of geometry

$$c = f\left(L_{chamber}, \frac{A_E}{A_T} \right)$$

By further refining this process, one could have a completely parametric engine design.

A simple way to obtain the performance curves could be to naively interpolate from specification of off the shelves components. While these will prove useful for the design process, it is crucial[31] to have the component design completely determined. That is, we need to be sure we can

- Manufacture the subsystem to meet the projected specification characteristics
- Produce the detailed design without human intervention.

Economic viability of Parametric design Vs Modular design
Parametric design could offer optimal performances while still taking advantage of scale economy.

Analysis of scale economy
Scale economy is based on the fact that fixed costs can be diluted over large volume of production. Fixed costs can be divided in

- **Design,Development and Testing**
  Obviously, design, development and testing costs have to be paid only once per product. With parametric design, instead, one would have to do at least non destructive tests on every component. However, the R&D costs would be spread out over a much wider application range.
- **Production Infrastructures**
  For very large volumes of production, these could include moulds and specified machinery. However, for aerospace purposes, these are mostly CNC machine and 3D printers. Hence, slightly modified designs will not affect the unit cost.
- **Material break cost**
  Due to large volume order, suppliers might offer a discount. For material this still applies for parametric design; for basic components such as integrated circuits, it doesn't.

Finally, an increase in the range of the module produced would widen the market, allowing for greater scale economy.

Optimal design
Assuming we have developed the parametric design for all the subsystems, we are still left with the embarrassing problem of the choice in $\mathbb{R}^n$. Once again, we can not define a total order because a generic system will use more than one resource to produce more than one good, therefore there is no unique definition of efficiency.

Using optimum management we can be sure that the system will always perform the best it can given the **system design** and **external conditions**. While the system design is what we need to find, we have not yet said anything about external conditions.

[31] For the economic worthiness of the endeavour

Application to a specific mission

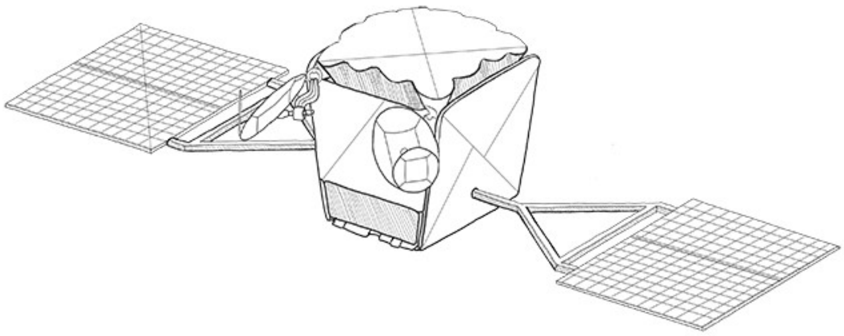


Figure 4.37: Oneweb concept for the *standard* nanosatellite.

In the previous chapters, we have developed a method for system design based on a *flexibility principle*, which can be also seen as simply relaxing the hypothesis of single controllable input subsystem. To assess the general proprieties and overall performances of this approach, it was applied to several thousand different mission scenarios, which were expressed by maximum output requirements (e.g. maximum power request, maximum computational burden etc...). Due to the varied nature of these mission, no optimization/target function was explicitly chosen and the only secondary goal was the minimization of all subsystem authorities. This was chosen under the assumption that any sensible cost function (such as total mass, total cost, complexity etch) is non decreasing with all subsystem authorities. Then, minimizing system authority will minimize any cost function.
Due to the wide variation of mission parameters, the results obtained are meaningful only on a mean, statistical base. Furthermore, a decrease in minimum needed authority

for a subsystem is hard to intuitively relate to a system improvement.
In order to increase our confidence in this approach, we will now apply it to a specific mission, with a specific optimization target. We have chosen to study a constellation of low orbit telecommunication satellites, with the secondary goal of mass minimization.

As explored extensively in chapter 3, the easiest way to achieve subsystem flexibility is by having two non identical components working together. The *flexibility approach* will then be achieved by choosing which components to use.

To measure the *effectiveness* of the flexibility method, we will compare it against both a classic, monolithic, design (in which we just want to chose the *single* best component) and a simply redundant, *fragmented* one, in which redundancy is achieved by using identical components. This second architecture is aimed at increasing system reliability while maintaining costs and design complexity by having identical components.
Since we have chosen to optimize for minimal total system mass, this will be our metrics to judge the effectiveness of a method.

Finally, since this is still a preliminary study, we will use simple (linear) models for the production function of each subsystem. At this stage, higher level of detail would not be relevant, as the uncertainties over mission definition and component model would hide any detailed system behaviour. Furthermore, quick and indicative models are often more useful than complex computational one, which would require a lot more time to process and would be much harder to understand intuitively. As for the model that links subsystem's authority to system's mass, we will initially assume this to be linear as well. However, under this linear hypothesis, it is clear that there will be no difference between monolithic design and the the fragmented one[32].

Mission Definition

We envision a constellation of a vast number ($\approx$ 700) of small satellites (mass $\approx 100 \sim 200$ kg) in low earth orbit ($h \approx 800 \sim 1200$ km) for telecommunication and internet providing purposes. Similar projects are being pursued by Oneweb, SpaceX and Surrey Satellite ltd, which makes it an interesting exercise both due to the richness of available information and for the intrinsic value of the study.

We will focus on the preliminary design of following subsystems: **Power system**, **ADCS**, **Telecom** (which, in this case, is almost the same as the payload) and **On board computer**.

[32] As the mass function is linear with authority, having two subsystems with half the authority is the same as having one subsystem

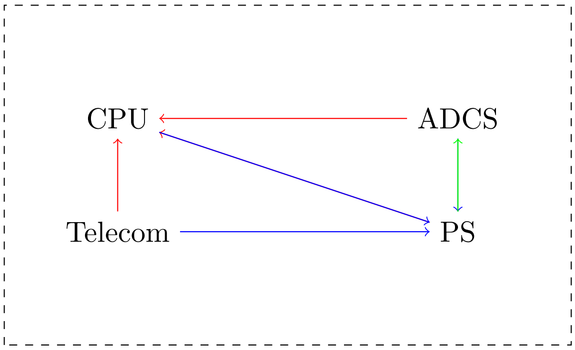


Figure 4.38: Map of the subsystem interdependencies.

We are going to use linear models for all the *production function* of each subsystem (i.e. all the goods/resources they consume or produce). Using linear models allows also for constant values, therefore the instantaneous production/consumption of a subsystem will be given by the consumption vector times the control variable ($\vec{V}_s \cdot x_s$) and a constant vector. The constant vector allows for a better representation of real system behaviour.

$$\text{CPU}: \begin{pmatrix} 0 \\ -i_C \\ +N_C \\ 0 \end{pmatrix} \cdot x_C \quad \text{ADCS}: \begin{pmatrix} T_A \\ -i_A \\ -N_A \\ 0 \end{pmatrix} \cdot x_A \quad \text{PS}: \begin{pmatrix} -T_P \\ +i_P \\ -N_P \\ 0 \end{pmatrix} \cdot x_P \quad \text{Telecom}: \begin{pmatrix} -T_T \\ -i_T \\ -N_T \\ D_T \end{pmatrix} \cdot x_T \tag{4.26}$$

The vectors $\vec{V}_s$ will change depending on the component we are considering and ideally they would be taken directly from the component's data sheet. All coefficient are intended to be real positive numbers.

While it is clear that the CPU needs power to provide the N_C computational power, other interactions might not be as obvious. Depending on the orbit or the spacecraft manoeuvres, the power system might need to perform sun tracking by rotating the solar panels. This is equivalent as an external torque which will have to be corrected by the ADCS. Furthermore, both sun tracking and power conversion algorithms will require some computational power. Finally, even the telecommunication system might use a pointing mechanism, which will again require adjustments from the ADCS, but this has not been included in the model.

Remark. The model we have implemented in the previous chapter featured a great number of parameters, but many of them were just definition of mission requirements. Since the mission parameters now will be fixed however, we can considerably reduce the number of variables.

Once we have chosen the subsystem from a catalogue, the whole system will be model by the following

$$\begin{pmatrix} 0 \\ -i_C \\ +N_C \\ 0 \end{pmatrix} \cdot x_C + \vec{C}_C + \begin{pmatrix} T_A \\ -i_A \\ -N_A \\ 0 \end{pmatrix} \cdot x_A + \vec{C}_A + \begin{pmatrix} -T_P \\ +i_P \\ -N_P \\ 0 \end{pmatrix} \cdot x_P + \vec{C}_P + \begin{pmatrix} -T_T \\ -i_T \\ -N_T \\ D_T \end{pmatrix} \cdot x_T + \vec{C}_T = \begin{pmatrix} T_{req}(t) \\ i_{req}(t) \\ N_{req}(t) \\ D_{req}(t) \end{pmatrix}$$

$$\begin{bmatrix} 0 & T_A & -T_P & 0 \\ -i_C & -i_A & +i_P & -i_T \\ +N_C & -N_A & -N_P & 0 \\ 0 & 0 & 0 & D_T \end{bmatrix} \cdot \begin{pmatrix} x_A \\ x_P \\ x_C \\ x_T \end{pmatrix} = \begin{pmatrix} T_{req}(t) \\ i_{req}(t) \\ N_{req}(t) \\ D_{req}(t) \end{pmatrix} - \sum \vec{C}_i \tag{4.27}$$

Where all the constant vector have been lumped together on the right side of the equation. If the system is able to provide the requested outputs without exceeding the maximum authority of each subsystem, mission requirements are met. This condition is easily expressed by

$$\vec{x} \ll \vec{Aut} \quad \text{where} \quad \vec{x} \doteq \mathbf{S}^{-1}(\vec{T}_{req} - \vec{C}), \quad \vec{Aut} = \begin{bmatrix} Aut_1 \\ Aut_2 \\ \dots \\ Aut_n \end{bmatrix}$$

Once we have verified that the system is able to meet mission requirements, we assess its effectiveness which in this case, is its mass. This is obtained by simply adding the each subsystem mass, which should be reported on the component data sheet.
Later in this chapter, instead of using actual components, we will use equations to model a continuous range of authorities. In that case, total system mass will become a function of each subsystem authority.

Mission requirements

We need to define the right hand side of equation 4.27, or the amount of each of the outputs we require for the specified mission. As a first approximation, we can limit our requests to only the attitude control and volume of data transmitted. This because we assume that no power output is requested, as all the power consumption from the system itself have already been accounted for. It would not be the case if we had divided the system in bus and payload, in which case, the bus would have had to supply the payload with everything it needed . Here, the payload is the telecommunication system, and therefore we can assume to have everything together.

To describe the mission with a series of request vectors, we need to define one for each situation that could saturate our actuators. These vectors are given by the maximum expected requests for each system, while the other requests are assumed to be average, which is justified under the assumption that request intensities are probabilistic and peak requests are unlikely. For each output we than have to identify average and maximum levels.

$$\text{Ext}_1 = \begin{bmatrix} T_{max} & T_{avg} \\ 0 & 0 \\ 0 & 0 \\ D_{mean} & D_{avg} \end{bmatrix} \qquad \text{Ext}_2 = \begin{bmatrix} T_{max} & T_{avg} & T_{avg} & T_{avg} \\ i_{avg} & i_{max} & i_{avg} & i_{avg} \\ N_{avg} & N_{avg} & N_{max} & N_{avg} \\ D_{mean} & D_{mean} & D_{mean} & D_{avg} \end{bmatrix}$$

Torque requirements

The torque required for any manoeuvre is given by $T_i = I_i\ddot{\theta}_i$. To assess it, we need to know both the desired time law $\theta(t)$, and the inertial proprieties of the spacecraft. To obtain the first, we need to chose a controller algorithm, while the second one will depend on the configuration we are examining. In the matlab algorithm, the inertia of the satellite is estimated using the mass, the volume (using cubesat's standard maximum density) and some assumption on the shape of the system. As theses parameters will be determined during the simulation, instead of giving maximum and average torque requirement, we will specify maximum and average angular acceleration.

We want to derive $\theta(t)$ as a function of simple mission requirement. These will be specified as a rotation of the angle $\Delta\theta$ in the time interval Δt. Since the objective is to find the maximum performance we can achieve with a given authority T, we will assume that we will be using the full authority for the whole time. Then the we have the curve in figure 4.39, which assumes no coasting and therefore

$$T_{RW} = \frac{4\Delta\alpha}{(\delta t)^2} \cdot I_{sat} \tag{4.28}$$

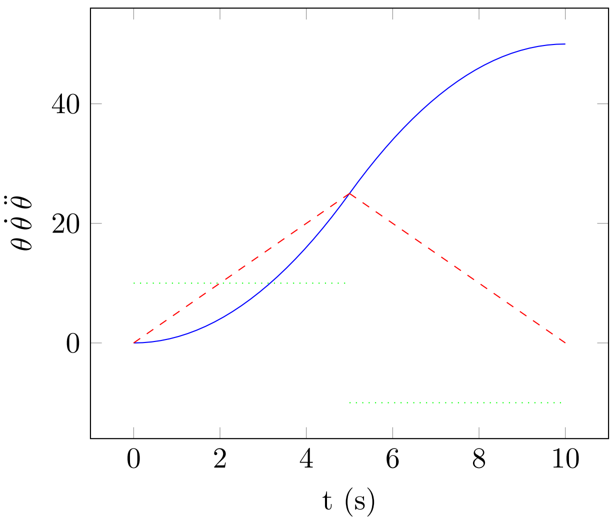


Figure 4.39: Angular acceleration, velocity and rotation.

Since the constellation is in LEO, we assume the period of the orbit to be approximately 90 minutes, which gives an average rotation (in nadir pointing configuration) of $360/5400\,[^\circ s^{-1}] \approx 1,2\,^\circ s^{-1}$. Therefore, assuming we do not relay on passive attitude control techniques, this is the rotational speed we need to maintain. Of course, if this is constant, we do not need to supply any torque. If we assume that the average nadir point active correction is to be executed every 10 seconds, and takes five, we have

$$\Delta\theta_{avg} = 12^\circ \qquad \Delta t_{avg} = 5\,\mathrm{s}$$

To estimate maximum torque requirement, we assume it to be there times as important as in the average case, hence

$$\Delta\theta_{avg} = 36^\circ \qquad \Delta t_{avg} = 5\,\mathrm{s}$$

Data requirement are hard to assess; by looking at telecommunication satellites specifications, traditional telecommunication satellite might have around 100 transponders. Approximately, this amounts to about 50 000 Mbps of total data link for one geostationary satellite. Since a typical GEO constellation might have tens of satellites, while the one we are considering might have approximately 1000, we estimate budget about 500 Mbps on average and 1 000 Mbps peak requirement per satellite.

Solar panel angular acceleration $\ddot{\theta}$ for power system estimation
Since we are assuming that the power system (in a worst case scenario of unusable battery) needs to have an independent orientation system from everything else, we assume it needs to counter act the motion of the satellite, in order to keep its pointing. Then the angular acceleration we expect will be $\ddot{\theta} \approx \ddot{\theta}_s$.

Derivation of Subsystems models

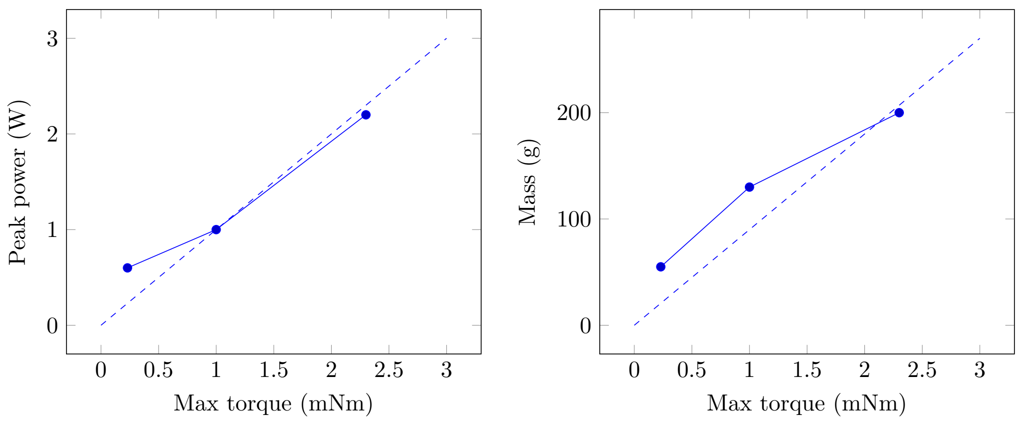
Ideally, we would like to implement the model shown in equation 4.26 directly, using only off the shelves components. To do so, we would need to compile a database containing the production function vector, system authority and system mass for each component. Given a selection of subsystems, we could then check weather the overall system is able to satisfy mission requirements, (which means that for each subsystem, the maximum authority needed to meet requirement is lower than that of the subsystem) and finally compute its total mass.
Once this process is performed for every possible combination of subsystems, we would be able to chose the best option according to our preferences (in this case, the system with lower mass).

However, this approach is not feasible due to the scarce number of standardize, *ready to buy*, components. Even after restricting the scope of the mission to cubesat-like components, by far the most standardized segment, one should not expect to find more than 10 different options for any given subsystem. Due to the poor resolution in the choice of subsystems, it is indeed possible that the difference between the standard design method and the flexibility based one might yield no effect.

To avoid this *filtering effect*, we will assume to have a continuous choice of subsystems, by allowing for any technology to be scaled up or down in a linear fashion with its authority. This is not as extravagant as it may sound; it can be seen as simply providing the requirements for each subsystem and then having it custom made.

To derive such models, we examine the components that are readily available on sites such as CubesatShop, Isis and Clyde Space and interpolate. An example for reaction wheel is provided below.

Example 4.25. Reaction wheel scalability
Cube Space offers three reaction wheels; small, medium and large. Their performances are shown in the table and the plot below



Cubewheel	Small	Medium	Large
Maximum Torque	0.23 mNm	1 mNm	2.3 mNm
Peak power	0.6 W	1.0 W	2.2 W
Mass	55 g	130 g	200 g

Figure 4.40: Cube Space small, medium and large reaction wheels.

We will assume that any reaction wheel with authority comparable with the above can be produced; all of the proprieties relative to its production function (e.g. power consumption, computational needs etc) will be model as a linear function of the torque output, while its mass (and possibly cost) will be a linear function of its maximum authority.

$$\text{RW} = \begin{pmatrix} 1 & & \\ -1.5 & [\frac{mW}{mNm}] & \\ -10 & & [\frac{MHz}{mNm}] \\ 0 & [\frac{Mbps}{mNm}] & \end{pmatrix} \cdot x\,[mNm] \qquad \text{mass RW} = 90[\frac{g}{mNm}] \cdot \max(x\,[mNm])$$

In the next paragraphs, we will extrapolate linear models for each subsystem. Some of the characteristics (for example the amount of computational power needed for a manoeuvre) are hard to estimate and are not usually included in the component data sheet. In these cases, rough estimation will be used and check against literature.

ADCS systems

The following table shows all the components we have been able to identify, with their basic specifications and proprieties.

Name RW	Authority [mNm]	Power consumption (peak) [mW]	Mass [g]	Price $
a*	0.23	600	55	4300
b**	0.635	2200	90	7100
c**	1	1000	130	5400
d*	2.3	2200	200	6500
10 SP-M	11	3500	960	165000
100 SP-0	110	10000	2600	258000
SSTL	240	140000	5200	NA
Clyde Space	40	28000	1500	34000

Figure 4.41: *: IsiSpace, ** cubesatshop.

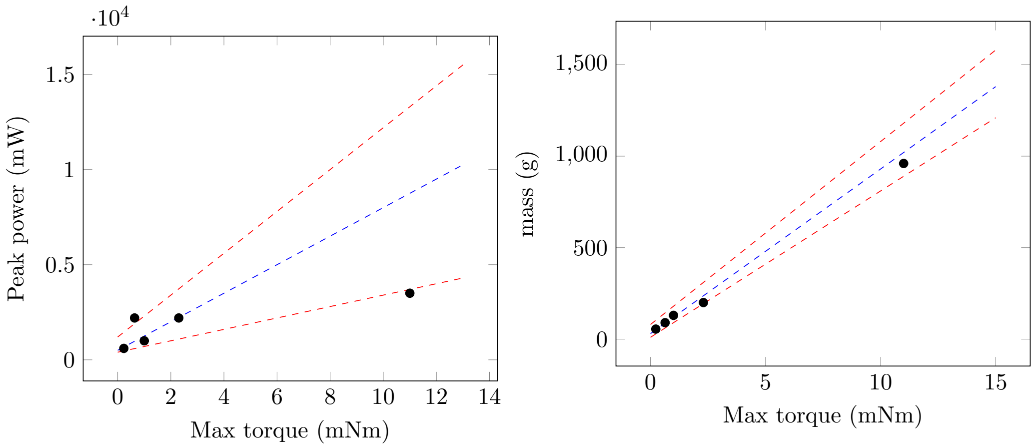


Figure 4.42: General trends for reaction wheels

From fig 4.42 we can abstract the model for both power consumption and mass. Since we want to have different components, we will not use the average proprieties of the cluster, but rather extract 3 trends, upper and lower bounds, and mean

$$\mathrm{RW}_A = \begin{bmatrix} 1 & & \\ -1.1\cdot 10^3 & \frac{mW}{mNm} & \\ -20 & & \frac{MHz}{mNm} \\ 0 & & \end{bmatrix} + \begin{bmatrix} 0 \\ -1200 \\ 0 \\ 0 \end{bmatrix} \qquad \mathrm{m}_{RW\,A} = 80\left[\frac{g}{mNm}\right]\cdot \text{ Autorithy } \mathrm{RW}_A{+}10 \tag{4.29}$$

$$\mathrm{RW}_B = \begin{bmatrix} 1 & & \\ -0.75\cdot 10^3 & \frac{mW}{mNm} & \\ -44 & & \frac{MHz}{mNm} \\ 0 & & \end{bmatrix} + \begin{bmatrix} 0 \\ -500 \\ 0 \\ 0 \end{bmatrix} \qquad \mathrm{m}_{RW\,B} = 90\left[\frac{g}{mNm}\right]\cdot \text{ Autorithy } \mathrm{RW}_B{+}30 \tag{4.30}$$

$$\mathrm{RW}_B = \begin{bmatrix} 1 & & \\ -0.3\cdot 10^3 & \frac{mW}{mNm} & \\ -110 & & \frac{MHz}{mNm} \\ 0 & & \end{bmatrix} + \begin{bmatrix} 0 \\ -400 \\ 0 \\ 0 \end{bmatrix} \qquad \mathrm{m}_{RW\,C} = 100\left[\frac{g}{mNm}\right]\cdot \text{ Autorithy } \mathrm{RW}_B{+}80 \tag{4.31}$$

To figure out a law for the computational power needed for our hypothetical parametric ADCS is based on ADCS computer boards.
Many ADCS computer board can be used as OBC as well however, very few data points are available to estimate the processor's speed. To have at least an indication of the processor's burden, we will assume it to be the only cause for energy consumption other that the actuators themselves. Then using the power consumption in the *idle* case, we can extrapolate the speed of the processors. The model we are using to link the two is based on the data from the SatBus 1C1

$$\text{Processor's Power consumption}[mW] = \begin{cases} 1.4 \\ 3.5 \\ 8 \end{cases} \left[\frac{mW}{MHz}\right] \cdot \text{ Processor's speed} \tag{4.32}$$

Board ADCS	Type	Authority [mNm]	Speed [MHz]	Power [mW]	Mass [g]	Cost [K$]	Inferred (at 3.5) [MHz]
Nano avionics	MT	/	168	NA	44	NA	/
MAI board	MT,RW	NA	NA	90	90	25	25
ISIS MT Board	MT	/	NA	175	196	8	50
CubeControl	MT, RW	0.15	NA	220	115	4.8	62
MAI 400	RW,MT	0.635	NA	450	694	90	128

The best attempt at modelling the computational burden of a set of 3 reaction wheels is by using the equation

$$\text{ADCS CPU}[MHz] = \begin{cases} 110 \\ 44 \\ 20 \end{cases} \frac{MHz}{mNm} \cdot T[mNm]$$

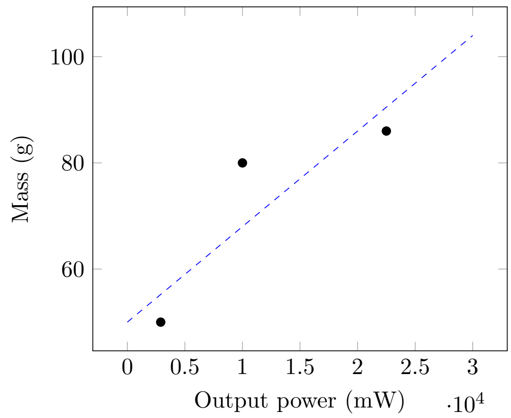
Power subsystem

The driving parameter in this subsystem is clearly the output current/wattage. This is usually obtained with a combination of direct solar power and battery usage. We will model the power system as 3 different components; **the distribution and control board(EPS), the solar array and the battery**. The mass of the board/distribution system will be estimated using table below and will be assumed to be

$$m_{board}[\text{g}] = 1.8 \cdot 10^{-3} \left[\frac{g}{mW}\right] \text{ Power } [mW] + 50$$

Name EPS	Max W [mW]	CPU [MHz]	Consumption [mW]	Mass [g]	Price [k$]
Crystalspace Vasik	10 000	NA	15	80	2.8
CubeSat DHV-CS-10	29 000	NA	NA	50	NA
CS 3rd Generation 1U EPS	22 500	NA	NA	86	4.9
Surrey PCDU	2 600 000	NA	NA	2600	NA

Figure 4.43: Power subsystems boards.



To complete the model as in equation 4.26, we need to link instantaneous power request to both torque and computational burden. The worst case scenario we wish to consider is that in which the battery is completely drained, and therefore we need to use the solar panels to provide all the needed wattage.
The solar array must be always pointed in the *appropriate direction*. By orienting them

we create a parasite torque, which we can estimate from the mass and geometry of the panels. Solar panels proprieties are reported below.

Name Solar Panel	Rated W [10^3 mW]	Eff [%]	Area	t [mm]	Mass [g]	Price [k$]	Dim [mm^2]	I [g mm^2]$\cdot 10^4$
CubeSat DHV-CS-10	2.9	30	1 U	2.4	50	NA	100x100	8.2
Nano avionics	2.3	17	1U	1.7	NA	NA	100x100	NA
ISIS single	3.6	30	1 U	2	50	NA	100x100	8.2
ISIS custom 1U	2.3	28	1 U	2	50	NA	100x100	8.2
ISIS custom 2U	4.6	28	2 U	2	100	NA	200x100	16.4
ISIS custom 3U	6.9	28	3 U	2	150	NA	300x100	24.6
EXA deployable a	3.75	19	3U	1.5	87	NA	300x100	14.5
EXA deployable b	12	28	4U	1.5	70	NA	400x100	11.6
EXA deployable c	3.75	19	3U	1.5	115	NA	300x100	19
EXA deployable d	12	28	4U	1.5	98	NA	400x100	16

Figure 4.45: Solar panels proprieties for inertia considerations

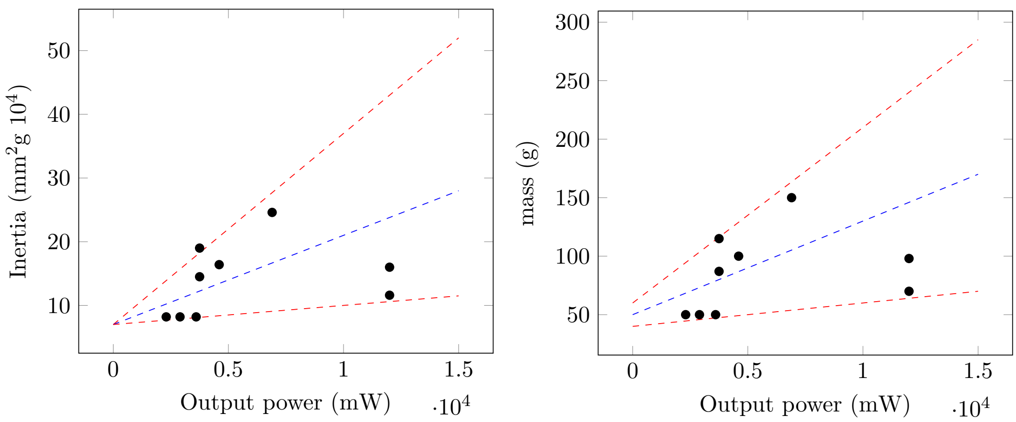


Figure 4.46: Linear models for solar arrays.

$$\text{Inertia Solar Panel } [10^4 \cdot mm^2 g] = 7 + \begin{cases} 3 \cdot 10^{-3} \\ 1.4 \cdot 10^{-3} \\ 0.3 \cdot 10^{-3} \end{cases} \left[\frac{mm^2 g}{mW}\right] \cdot \text{Output power}[mW]$$

Finally, we need to evaluate the computational burden of both the Peak Power tracking and solar tracking algorithm. From table 4.43 we could infer (using the same method used for the ADCS) a single data point at around 4.2 MHz [33] for the Crystalspace Vasik.

$$\mathrm{PS}_A = \begin{bmatrix} -(7+3\cdot 10^{-3}\mathrm{Aut})\cdot 10^{-2}\cdot \ddot{\theta} & \frac{mNm}{mW} \\ 1 & \\ \mathbf{-1.8\cdot 10^{-4}} & \frac{Mhz}{mW} \\ 0 & \end{bmatrix} \cdot W_{PS}$$

$$\mathrm{m}_{PS\,A}\,[g] = (15{\cdot}10^{-3}{+}1.8{\cdot}10^{-3})\left[\frac{g}{mW}\right]\cdot \text{ Autorithy } \mathrm{PS}_A{+}(60{+}50)g{+}8{\cdot}10^{-3}\text{Capacity battery}$$

$$\mathrm{PS}_B = \begin{bmatrix} -(7+1.4\cdot 10^{-3}\mathrm{Aut})\cdot 10^{-2}\cdot \ddot{\theta} & \frac{mNm}{mW} \\ 1 & \\ \mathbf{-4.2\cdot 10^{-4}} & \frac{Mhz}{mW} \\ 0 & \end{bmatrix} \cdot W_{PS}$$

$$\mathrm{m}_{PS\,B}\,[g] = (8{\cdot}10^{-3}{+}1.8{\cdot}10^{-3})\left[\frac{g}{mW}\right]\cdot \text{ Autorithy } \mathrm{PS}_B{+}(50{+}50)g{+}8{\cdot}10^{-3}\text{Capacity battery}$$

$$\mathrm{PS}_C = \begin{bmatrix} -(7+0.3\cdot 10^{-3}\mathrm{Aut})\cdot 10^{-2}\cdot \ddot{\theta} & \frac{mNm}{mW} \\ 1 & \\ \mathbf{-10.7\cdot 10^{-4}} & \frac{Mhz}{mW} \\ 0 & \end{bmatrix} \cdot W_{PS}$$

$$\mathrm{m}_{PS\,C}\,[g] = (2{\cdot}10^{-3}{+}1.8{\cdot}10^{-3})\left[\frac{g}{mW}\right]\cdot \text{ Autorithy } \mathrm{PS}_C{+}(40{+}50)g{+}8{\cdot}10^{-3}\text{Capacity battery}$$

Finally, the mass of the battery will be assumed linear with its capacity. Below, some target batteries are reported.

[33] or 1.8 or 10.7, depending on the computer model you use

Name	Voltage [V]	Ahr [mAh]	Capacity [mWhr]	Mass [g]	Price [k$]
Tenergy RCR123A cells	3	900	2700	17	0.008
Panasonic CGR18650HG cells	3.7	2250	8325	45	0.01
Clyde Space Remote battery board	8.2	1220	10004	62	1
Clyde Space 3U CubeSat Battery	8.2	1220	10004	141	1
CubeSatKit / Pumpkin CubeSat Kit Linear EPS Rev D Standard	3.7	3000	11100	155	0.750
GomSpace - NanoPower BP-4	8.4	3600	30240	213	1.5

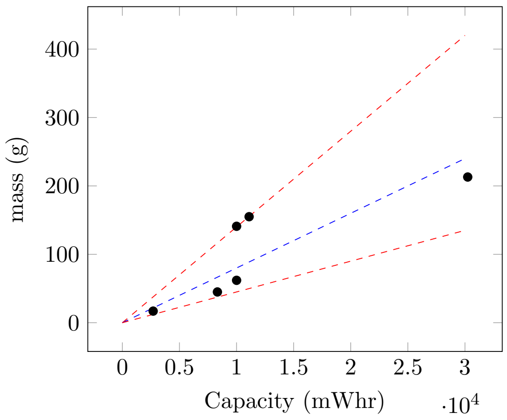


Figure 4.47: Battery mass Vs capacity

$$\text{mass battery}[g] = \begin{cases} 14 \cdot 10^{-3} \\ 8 \cdot 10^{-3} \\ 4.5 \cdot 10^{-3} \end{cases} \left[\frac{g}{mWhr}\right] \cdot \text{battery capacity}[mWhr]$$

Unlike the solar array and the power management board, the *authority* for the battery is its capacity. To estimate it, we will take the average power consumption and integrate it over the eclipse time.

$$\text{Battery capacity} = \int_{Eclipse} \text{avg}(mW)dt$$

Where average consumption will be obtained by imposing as external condition the average request for all subsystems.

Remark. We calculated inertia of the solar panels as $g\,mm^2$ but we express torque as mNm. The conversion between the two is

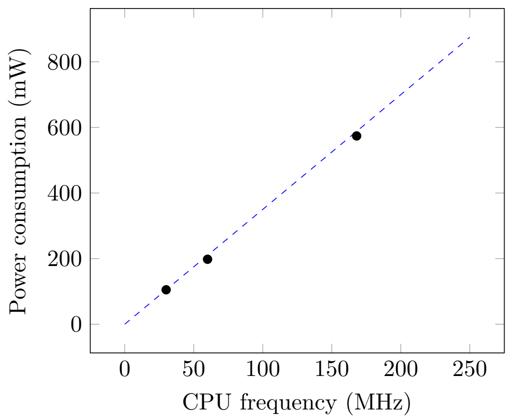
$$g\,mm^2 = 10^{-3}kg\,10^{-6}m^2 = 10^{-9}Nm \cdot s^2 = 10^{-6}mNm \cdot s^2 \quad 10^4 g\,mm^2 = 10^{-2}mNm\,s^2$$

On Board computer

For the OBC, the main driver will be the rate of the processor and the power consumption. Other characteristics are going to be included as well for future reference.

Board	Processor MHz	Max Power consumption mW	Mass g	Price $
ISIS OBC	400	550	94	4300
Cube computer	48	435	70	4500
SatBus 1C1	168	574	44	NA
Clyde Space OBC	100	350	NA	8400

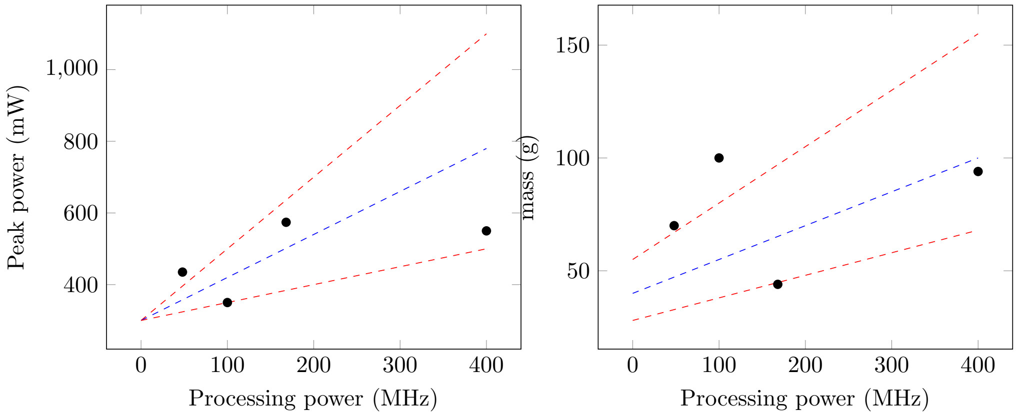
Sadly there are incredibly few OBC to choose from. From the data sheet of the SatBus 1C0, we can validate the linear model to infer correlation between number of operations and the amount of power consumed.



	30 MHz	60 MHz	168 MHz
current (2x 3.3 V)	16 mA	30 mA	87 mA
Power	105 mW	198 mW	574 mW

Figure 4.48: Linear relationship between clock frequency and power consumption in the SatBus 1C1

To scale the OBC regarding the relationships processor-power consumption and processor-mass, we present the following plots, which are extremely effective in conveying the oversimplification of the linear assumption.



$$\mathrm{OBC}_A = \begin{bmatrix} 0 & \\ -2 & \frac{mW}{MHz} \\ 1 & \\ 0 & \end{bmatrix} \cdot N_{OBC} + \begin{bmatrix} 0 \\ -300 \\ 0 \\ 0 \end{bmatrix} \qquad \mathrm{m}_{OBC\,A}\,[g] = 0.25 \left[\frac{g}{MHz}\right] \cdot \text{ Autorithy } \mathrm{OBC}_A + 55 \tag{4.33}$$

$$\mathrm{OBC}_B = \begin{bmatrix} 0 & \\ -1.2 & \frac{mW}{MHz} \\ 1 & \\ 0 & \end{bmatrix} \cdot N_{OBC} + \begin{bmatrix} 0 \\ -300 \\ 0 \\ 0 \end{bmatrix} \qquad \mathrm{m}_{OBC\,B}\,[g] = 0.15 \left[\frac{g}{MHz}\right] \cdot \text{ Autorithy } \mathrm{OBC}_B + 40 \tag{4.34}$$

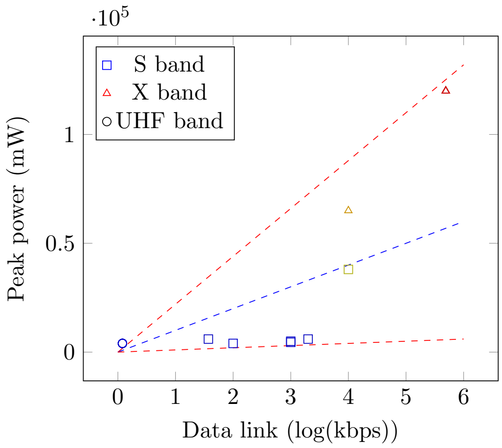
$$\mathrm{OBC}_C = \begin{bmatrix} 0 & \\ -0.5 & \frac{mW}{MHz} \\ 1 & \\ 0 & \end{bmatrix} \cdot N_{OBC} + \begin{bmatrix} 0 \\ -300 \\ 0 \\ 0 \end{bmatrix} \qquad \mathrm{m}_{OBC\,C}\,[g] = 0.1 \left[\frac{g}{MHz}\right] \cdot \text{ Autorithy } \mathrm{OBC}_C + 28 \tag{4.35}$$

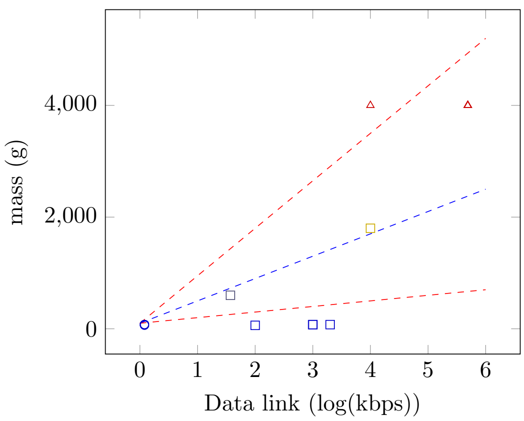
Telecom subsystem

There are several uplink downlink parameters, the most relevant to our model are reported in the table below.

Name	Type	Data rate	Signal power [mW]	Consumption [mW]	Mass [g]	Price $
a	S band	38 kbps	250	6000	600	NA
b	S band	10 Mbps	4000	38000	1800	NA
e**	S band	100 kbps	500	4000	62	8500
f**	S band	1 Mbps	500	5000	75	7800
Nano avionics	S band	1 Mbps	500	4500	75	NA
Clyde Space	S band	2 Mbps	500	6000	?	8900
c	X band	10 Mbps	5000	65000	4000	NA
d	X band	500 Mbps	11000	120000	4000	NA
g**	UHF/ VHF	1.2 kbps	500	4000	65	8500
h**	UHF/ VHF	1.2 kbps	500	4000	75	8500

Figure 4.50: Surrey, *: IsiSpace, ** Cubesatshop.





To asses the amount of computational power required we will assume that the number of operation performed by the CPU is proportional to volume of data to be transmitted.

$$\frac{MHz}{kbps} = \frac{1000\,mW}{1.8 \text{ or } 3.5 \text{ or } 8\frac{mW}{MHz}} \cdot \frac{1}{log_{10}(500000kbps)} = \frac{1000\,mW}{1.8 \text{ or } 3.5 \text{ or } 8} \cdot \frac{1}{5.6990} = \begin{cases} 97.4830 \\ 50.1341 \\ 21.9337 \end{cases}$$

$$\text{Tel}_A = \begin{bmatrix} 0 & & \\ -2.2 \cdot 10^4 & \frac{mW}{kbps} & \\ \mathbf{-22} & & \frac{MHz}{kbps} \\ 1 & & \end{bmatrix} \cdot \text{Log(Data)} \quad m_{Tel\,A}\,[g] = 100\left[\frac{g}{kbps}\right] \cdot \text{ Autorithy Tel}_A{+}100$$

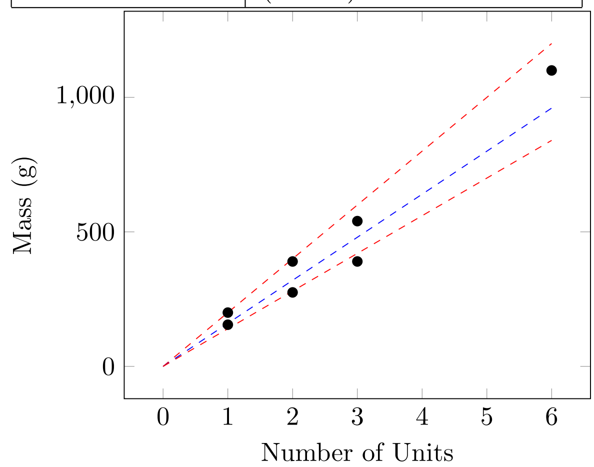
$$\mathrm{Tel}_B = \begin{bmatrix} 0 & & \\ -10^4 & \frac{mW}{kbps} & \\ \mathbf{-50} & & \frac{MHz}{kbps} \\ 1 & & \end{bmatrix} \cdot \mathrm{Log(Data)} \quad \mathrm{m}_{Tel\,B}\,[g] = 450\left[\frac{g}{kbps}\right] \cdot \text{ Autorithy } \mathrm{Tel}_B + 100$$

$$\mathrm{Tel}_C = \begin{bmatrix} 0 & & \\ -10^3 & \frac{mW}{kbps} & \\ \mathbf{-97} & & \frac{MHz}{kbps} \\ 1 & & \end{bmatrix} \cdot \mathrm{Log(Data)} \quad \mathrm{m}_{Tel\,C}\,[g] = 850\left[\frac{g}{kbps}\right] \cdot \text{ Autorithy } \mathrm{Tel}_C + 100$$

Structure mass

Finally, to have a more accurate representation of the satellite mass, we need to account for the structure. From the first estimation of the system mass we can obtain minimum volume (using maximum density, which for cubesat standards is 1 kg per litre).
Once we have an indication on the number of modules, we use the table below to asses the average mass for a 1 U structure.

Name	Type	Mass [g]	Cost [k$]
Clyde Space 1U	1 U	155	4.6
Clyde Space 2U	2 U	275	5.3
Clyde Space 3U	3 U	390	6.9
ISIS 1U	1U	200	2.15
ISIS 2U	2U	390	2.95
ISIS 3U	3U	540	3.65
ISIS 6 U	(2x3x1)U	1100	7.35



$$\text{Mass structure} = \begin{cases} 200 \\ 160 \\ 140 \end{cases} \left[\frac{g}{\#U} \right] \cdot \text{Number of units}$$

Algorithm structure

We now provide an overview of the algorithm that implements the classical approach and the one we propose. As the torque required of the reaction wheels depends on the inertia of the system, which is initially unknown, the process requires iterations, as shown in Fig. 4.52.

Give a set of all the subsystem types we could use, we can generate the list of every possible combination of subsystem. We want to assign to each one its mass, in order to find the lightest system. Since we are using parametric models, the mass of each component will be a function of its authority. To derive the authority of each subsystem we use the worst external conditions (for each subsystem), which guarantees that we satisfy the mission requirement.

As noticed above, some requirements depend on system mass. An estimation m_0 starts the algorithm. To obtain the inertia, we assume maximum density for a cubesat module to be 1000 kg m^{-3}, which we use to calculate the volume and therefore the number of units. Assuming a simple configuration of stacked cubes, we have everything we need to determine the moment of inertia.

With the satellite inertia we can express mission requirements as system requirements. Due to system coupling or interdependencies, it might not be intuitive which condition is the most demanding for a specific subsystem. Therefore all system requirements are considered and subsystem authority is set using the maximum value.

Having defined every authority, we can calculate each component mass. Furthermore, we add the mass of the structure (linear with the number of modules) and the mass of the battery (linear with battery capacity). If the final mass is sufficiently close to the original estimate, we proceed to the next system configuration, otherwise we iterate the process.

Static Model

In the static case, the system is completely determined, as there are no redundancies. Therefore, any subsystem authority needed for a specific system requirement can be obtained by simply inverting the system matrix.

$$\mathbf{S} \cdot \vec{x} = \vec{R}_i \quad \Rightarrow \quad \vec{x}_i = \mathbf{S}^{-1} \vec{R}_i$$

After all conditions have been examined, system authority is set by

$$\text{Aut}_j = \max \{ \vec{x}_1 \cdot \hat{e}_j, \vec{x_2} \cdot \hat{e}_j, \dots \vec{x}_n \cdot \hat{e}_j, \}$$

Where $\hat{e}_j$ is the vector that extracts the j th component[34]

[34] $\hat{e}_j = (0, 0, .., 0, 1, 0, ..., 0)$

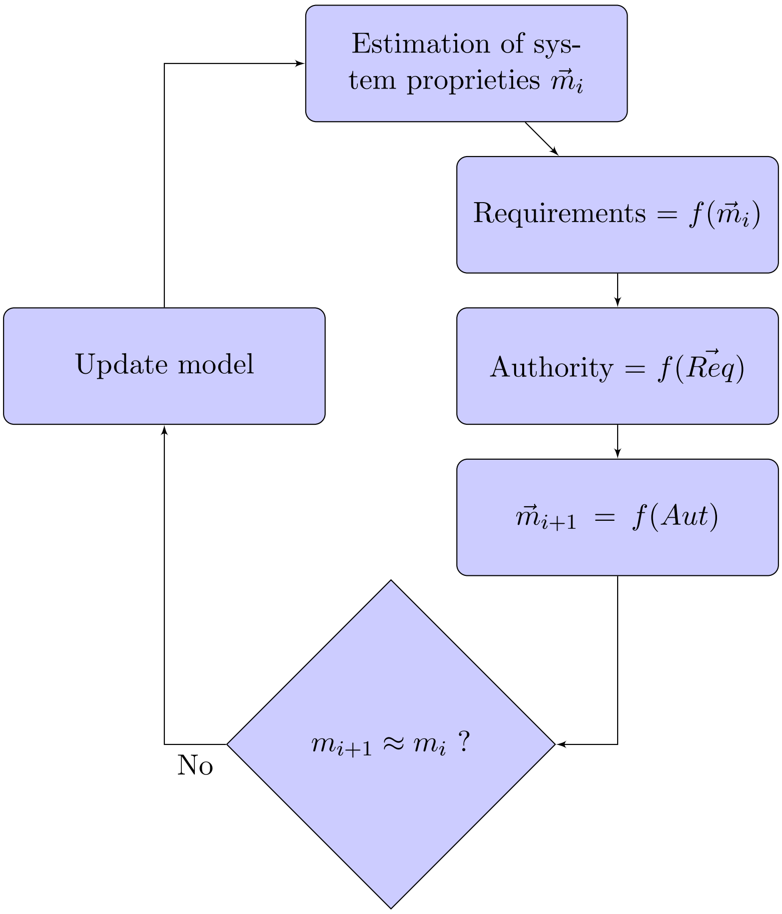


Figure 4.52: Steps in the algorithm.

Flexible Model

As we have chosen to pursue flexibility using redundancy, to use the flexibility approach we need to specify two different components for each subsystem. Mass estimation and system requirements can be derived exactly as in the previous case; system authority can not.

As shown in Fig. 4.53, the main difference between the two approaches is whether we have freedom in choosing the authority or not. As the system is doubly redundant in all of its components, we can model it as two separate system working together. Let us call $\hat{x}$ the vector of the operational level of the first set of actuators and $\hat{y}$ the operational level of the second one. $\mathbf{S_1}$ and $\mathbf{S_2}$ are respective matrices, while $\hat{E}_i$ is the ith external request.
If the system meets mission requirements, it must be that

$$[\mathbf{S_1}] \cdot \hat{x} + [\mathbf{S_2}] \cdot \hat{y} \leq \hat{E}_i \quad \forall i \in \text{External conditions}$$

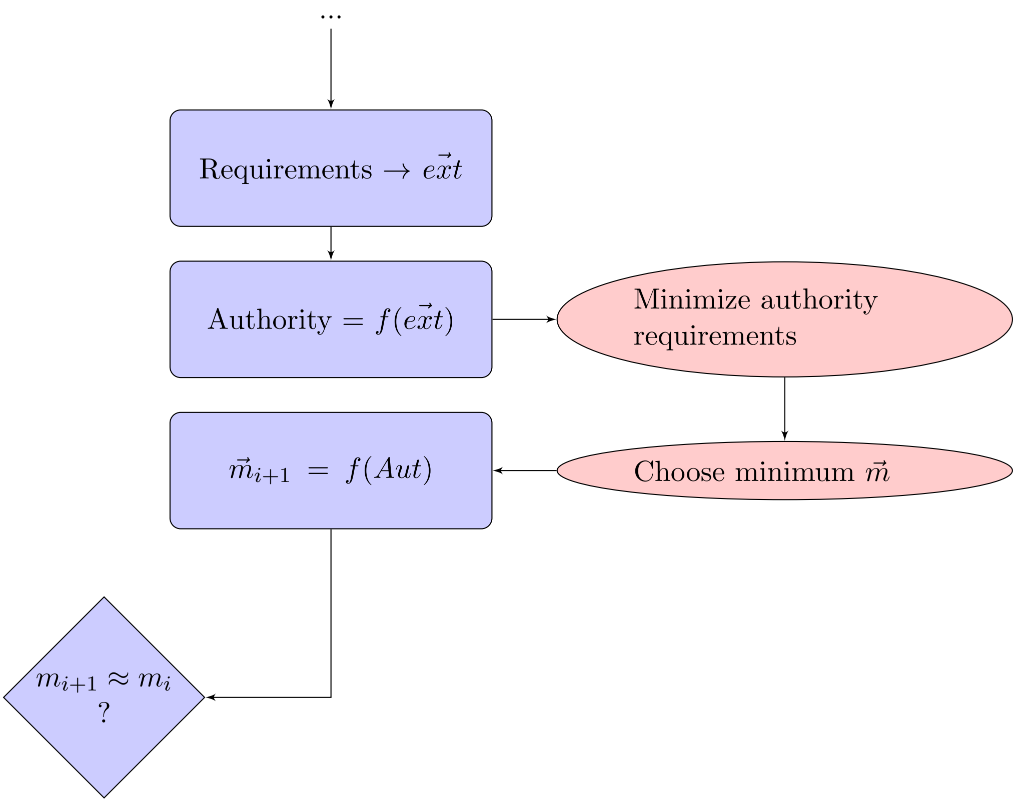


Figure 4.53: Differences in the pseudo code for system design with flexibility.

On the other hand, the mass of the system (which we wish to minimize) is given by

$$M_s = \hat{m}_1 \cdot \hat{x} + \hat{m}_2 \cdot \hat{y}$$

Where $\hat{m}_1$ and $\hat{m}_2$ are vector that depend on the choice of subsystems but are both strictly positive, which reminds us that we are seeking to minimize system authority. Clearly, if we had only one external condition, the lowest authority is given by

$$[\mathbf{S_1}] \cdot \hat{x} + [\mathbf{S_2}] \cdot \hat{y} = \hat{E}_1 \quad \Rightarrow \quad \hat{x} = [\mathbf{S_1}]^{-1}(\hat{E}_1 - [\mathbf{S_2}] \cdot \hat{y})$$

Hence

$$M_s = \hat{m}_1 \cdot [\mathbf{S_1}]^{-1} \cdot \hat{E}_1 + (-\hat{m}_1 \cdot [\mathbf{S_1}]^{-1} \cdot [\mathbf{S_2}] + \hat{m}_2)\hat{y}$$

Which is a function only of the authority of the secondary subsystems. Calling $\mathbf{S} \doteq [\mathbf{S_1}]^{-1} \cdot [\mathbf{S_2}]$, if we derive it with respect to each secondary subsystem authority we have

$$\frac{\partial M_s}{\partial y_i} = -\hat{m}_1 \cdot \mathbf{S}(:, \mathbf{i}) + \hat{m}_2(i)$$

If the derivative is positive, system mass increases as the more authority is given to the secondary system, if negative mass decreases.
Note two important proprieties for this model;

1. Each derivative is constant.

2. External condition don't affect the derivative.

Therefore minimum system mass, **regardless of the external conditions** is achieved by transferring all the authority to either the primary subsystem ($\frac{\partial M_s}{\partial y_i} > 0$, $y_i = 0$) or to the secondary ($\frac{\partial M_s}{\partial y_i} < 0 \Rightarrow y_i = y_{max}, x_i = 0$).

However, an autonomous system will usually have to react to more than one external condition. To account for this, we include all external condition vectors in the matrix $\mathbf{E} = [\hat{E}_1, \hat{E}_2, \dots, \hat{E}_s]$, where s is the number of external conditions. System behaviour in each condition can be written in matrix form as

$$[\mathbf{S_1}] \cdot [\hat{x}_1, \hat{x}_2, \dots, \hat{x}_s] + [\mathbf{S_2}] \cdot [\hat{y}_1, \hat{y}_2, \dots, \hat{y}_s] = \mathbf{E}$$

Which can be use to express the primary subsystem requirements as a function of the authority of the second one

$$[\hat{x}_1, \hat{x}_2, \dots, \hat{x}_s] = [\mathbf{S_1}]^{-1}(\mathbf{E} - [\mathbf{S_2}] \cdot [\hat{y}_1, \hat{y}_2, \dots, \hat{y}_s])$$

System mass is a function of authority, which needs can be written as

$$\vec{\mathrm{Aut}}_X = \begin{pmatrix} \max([x_1(1), x_2(1), \dots, x_s(1)]) \\ \max([x_1(2), x_2(2), \dots, x_s(2)]) \\ \dots \\ \max([x_1(n), x_2(n), \dots, x_s(n)]) \end{pmatrix} \qquad \vec{\mathrm{Aut}}_Y = \begin{pmatrix} \max([y_1(1), \dots, y_s(1)]) \\ \max([y_1(2), \dots, y_s(2)]) \\ \dots \\ \max([y_1(n), , \dots, y_s(n)]) \end{pmatrix}$$

We are interested in mass, and for each each subsystem we have

$$\begin{bmatrix} m_{ss\,1} \\ m_{ss\,2} \\ \dots \\ m_{ss\,s} \end{bmatrix} = \begin{bmatrix} \vec{m}_1(1) & 0 & \dots & 0 \\ 0 & \vec{m}_1(2) & \dots & 0 \\ & \dots & \dots & \\ 0 & 0 & \dots & \vec{m}_1(n) \end{bmatrix} \vec{Aut}_X + \begin{bmatrix} \vec{m}_2(1) & 0 & \dots & 0 \\ 0 & \vec{m}_2(2) & \dots & 0 \\ & \dots & \dots & \\ 0 & 0 & \dots & \vec{m}_2(n) \end{bmatrix} \vec{Aut}_Y \tag{4.36}$$

Clearly, we can express $\vec{Aut}_X$ as a function of $\vec{Aut}_Y$ and then optimize. However, the function max is not continuous. Therefore, we suggest to first calculate each subsystem mass and apply the maximum only afterwards.

We call $\mathbf{M}_1$ and $\mathbf{M}_2 \in \mathbb{R}^{n \times n}$ the matrices with the coefficients of mass on the diagonal (as used in eq. 4.36).

$$\mathbf{M}_s = \mathbf{M}_1 \cdot X + \mathbf{M}_2 \cdot Y = \mathbf{M}_1 \cdot [\mathbf{S_1}]^{-1}(\mathbf{E} - [\mathbf{S_2}] \cdot Y) + \mathbf{M}_2 \cdot Y$$

$$\mathbf{M}_s = \mathbf{M}_1 \cdot [\mathbf{S_1}]^{-1}\mathbf{E} + (\mathbf{M}_2 - \mathbf{M}_1 \cdot [\mathbf{S_1}]^{-1}[\mathbf{S_2}]) \cdot Y = \mathbf{A} + \mathbf{B} \cdot Y$$

The mass of the system is now given by the sum of the maximum values of mass for every row. The variable that we can use to change the matrix $\mathbf{M_s}$ are the $n \times s$ components of the Y matrix. Then, we want to study the derivative of total system mass M_{tot} with respect to these variables

$$\frac{\mathrm{d}M_{tot}}{\mathrm{d}\mathbf{Y}} = \frac{\mathrm{d}}{\mathrm{d}\mathbf{Y}}(m_{sys,1} + m_{sys,2} + \cdots + m_{sys,n}) = \frac{\mathrm{d}}{\mathrm{d}\mathbf{Y}}m_{sys,1} + \frac{\mathrm{d}}{\mathrm{d}\mathbf{Y}}m_{sys,2} + \cdots + \frac{\mathrm{d}}{\mathrm{d}\mathbf{Y}}m_{sys,n}$$

Since everything is linear, the derivative of M_{tot} with respect to $\mathbf{Y}$ will be the sum of the derivative of each subsystem mass; we can divide the problem by subsystem for simplicity.

Let us consider the minimization of $m_{sys,j}$ $(0 < j < n)$. This is the maximum value on the jth row of $\mathbf{M}_s$, which we assume to be on the ith column $(0 < i < s)$. Clearly, the matrix A plays no role as it is constant,

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{Y}}(\hat{e}_j^T \cdot \mathbf{M}_s \cdot \hat{e}_i) = \frac{\mathrm{d}}{\mathrm{d}\mathbf{Y}}\left(\begin{pmatrix} 0_1 \\ \dots \\ 1_j \\ \dots \\ 0_n \end{pmatrix}^T \begin{bmatrix} B_{1,1} & B_{1,2} & \dots & B_{1,n} \\ B_{2,1} & B_{2,2} & \dots & B_{2,n} \\ \dots & \dots & \dots & \dots \\ B_{n,1} & B_{n,2} & \dots & B_{n,n} \end{bmatrix} \cdot \begin{bmatrix} y_{1,1} & y_{1,2} & \dots & y_{1,s} \\ y_{2,1} & y_{2,2} & \dots & y_{2,s} \\ \dots & \dots & \dots & \dots \\ y_{n,1} & y_{n,2} & \dots & y_{n,s} \end{bmatrix} \begin{pmatrix} 0_1 \\ \dots \\ 1_i \\ \dots \\ 0_n \end{pmatrix}\right)$$

$$\frac{\mathrm{d}\ m_{sys,j}}{\mathrm{d}\mathbf{Y}} = \frac{\mathrm{d}}{\mathrm{d}\mathbf{Y}}\left(\begin{bmatrix} B_{j,1} & B_{j,2} & \dots & B_{j,n} \end{bmatrix} \cdot \begin{bmatrix} y_{1,i} \\ y_{2,i} \\ \dots \\ y_{n,i} \end{bmatrix}\right) = \begin{array}{c} \begin{matrix} 1 & \dots & i & \dots & s \end{matrix} \\ \begin{bmatrix} 0 & \dots & B_{j,1} & \dots & 0 \\ 0 & \dots & B_{j,2} & \dots & 0 \\ & \dots & \dots & & \\ 0 & \dots & B_{j,n} & \dots & 0 \end{bmatrix} \end{array}$$

If we now sum all derivatives, we have a matrix that gives us clear indications on whether to increase the authority of the primary or secondary subsystem. Every element element of the matrix $\frac{\mathrm{d}M_{tot}}{\mathrm{d}\mathbf{Y}}$ indicates if M_{tot} increases, decreases or is unaffected by the respective $y_{i,j}$.

$$\frac{\mathrm{d}M_{tot}}{\mathrm{d}\mathbf{Y}} = \sum_{j=1}^{n} \frac{\mathrm{d}\ m_{sys,j}}{\mathrm{d}\mathbf{Y}} \doteq \mathbf{D} \in \mathbb{R}^{n \times s} \tag{4.37}$$

Before we going into more detail on the optimization method, we need to consider that is the objective function is linear with authority(as in this case), the $\frac{\mathrm{d}\ M_s}{\mathrm{d}\mathbf{Y}}$ matrix is **piece wise** constant.

Locally, if $\mathbf{D}_{s,t} > 0$, mass increases with $y_{s,t}$, and vice versa. Many coefficient in $\mathbf{D}$ might also be zero, which means that they act on a condition that is not a maximum, and hence does not affect the authority of the subsystem.
As the coefficients seems to be independent of Y, and the derivative is constant, the extreme values are found at the extreme of the interval, either for y maximum or minimum. Clearly, the minimum value for any $Y_{i,j}$ is zero (all authority to the primary system), while the highest is that which yields $X_{i,j} = 0$(all authority to the secondary).

However, as we act on the matrix Y, $\mathbf{M}_s$ changes, and the position of the maximum value for each subsystem may change position in $\mathbf{M}_s$, which means that $\mathbf{D}$ changes as well.

Defining a piece wise domain is conceptually simple, at least in the linear case, however it may be tricky to generalize for a high number of dimension.

Problem with optimization is that the derivative is not defined on some points of the domain, where more than one system are at the maximum value. Therefore we decided to have and algorithm that optimize only within the continuous piece of the domain, and is randomly initiated at different points.

A much easier alternative is to implement an iterative algorithm.

Remark. There are some things to consider before proceeding onto the non linear case

1. External conditions do not affect the $\mathbf{D}$ matrix *directly*, however they define the domain of each piece of derivative.

2. More than one maximum might appear at once.

3. In this chapter, we are assuming that minimizing authority minimizes system mass, which is reasonable as each subsystem mass is strictly increasing with authority. However, we have neglected the effect that changing the ratio of authorities between the primary and secondary system might have on the control allocation effectiveness. It is not impossible that this approach might require bigger tanks or batter capacity to compensate for an eventual decrease in efficiency of the management algorithm.

   The interactions mentioned above are just possibilities, not the result of any empirical evidence. To assess such issue we would need to develop a much more sophisticated model, which would not be justified at this stage.

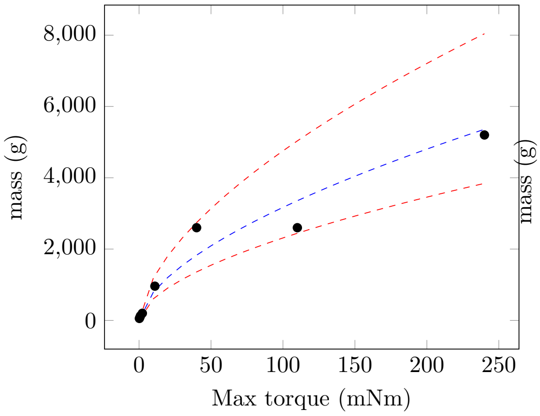
Using a non linear fit for system mass

Finally, we try to generalize the above method for non linear target functions as well. This allows for a more realistic representation of parametric design, but makes the problem significantly more complex. We are still assuming linear system behaviour.

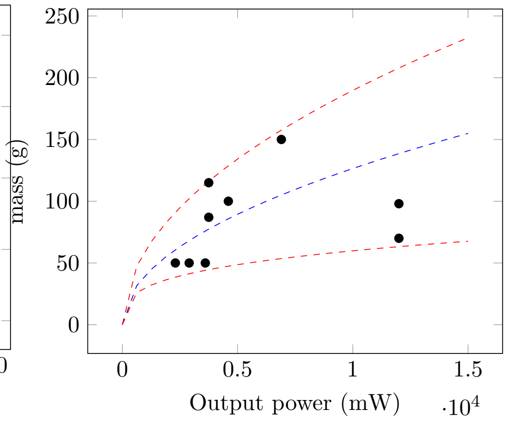
To begin with, we derive non linear equation to predict component mass as a function of authority. Since we are using the same database as in the linear case, which is very limited, we choose to adopt a function as simple as possible

$$m_s = C_1 \cdot Aut^{C_2} \tag{4.38}$$

where both constants are derived by interpolation. As we expect the cost function to be strictly increasing, both coefficients will be greater than zero. The results are listed below.



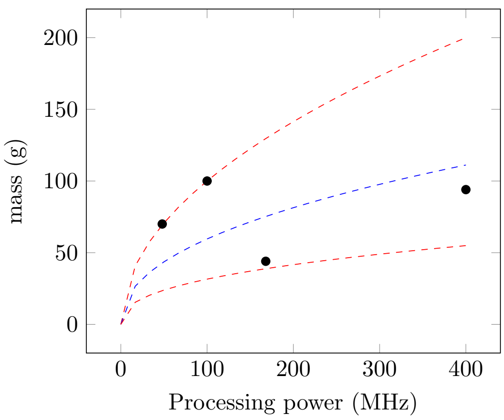
(a) Reaction wheels



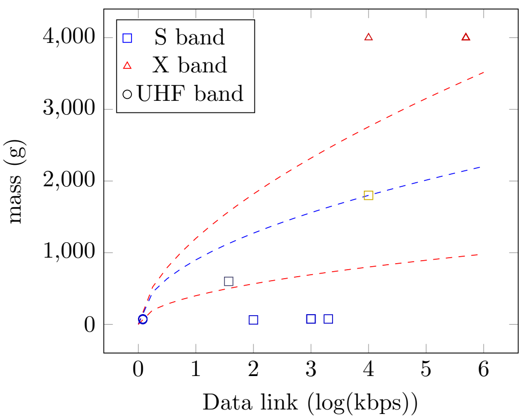
(b) Solar panel

$$\begin{cases} m_{RW\,A} &= 160 \cdot (\mathrm{Aut}_{RW})^{0.58} & [g] \\ m_{RW\,B} &= 200 \cdot (\mathrm{Aut}_{RW})^{0.6} & [g] \\ m_{RW\,C} &= 300 \cdot (\mathrm{Aut}_{RW})^{0.6} & [g] \end{cases}$$

$$\begin{cases} m_{PS\,A} &= 60 \cdot (\mathrm{Aut}_{PS} \cdot 10^{-3})^{0.5} & +1.8 \cdot \mathrm{Aut}_{PS} \cdot 10^{-3} & +m_{battery} \\ m_{PS\,B} &= 40 \cdot (\mathrm{Aut}_{PS} \cdot 10^{-3})^{0.5} & +1.8 \cdot \mathrm{Aut}_{PS} \cdot 10^{-3} & +m_{battery} \\ m_{PS\,C} &= 30 \cdot (\mathrm{Aut}_{PS} \cdot 10^{-3})^{0.3} & +1.8 \cdot \mathrm{Aut}_{PS} \cdot 10^{-3} & +m_{battery} \end{cases}$$



(a) On board computer



(b) Telecommunication system

$$\begin{cases} m_{OBC\,A} &= 10 \cdot (\mathrm{Aut}_{OBC})^{0.5} & [g] \\ m_{OBC\,B} &= 7.5 \cdot (\mathrm{Aut}_{OBC})^{0.45} & [g] \\ m_{OBC\,C} &= 5 \cdot (\mathrm{Aut}_{OBC})^{0.4} & [g] \end{cases}$$

$$\begin{cases} m_{TEL\,A} &= 400 \cdot (\log(\mathrm{Aut}_{TEL}))^{0.5} & [g] \\ m_{TEL\,B} &= 900 \cdot (\log(\mathrm{Aut}_{TEL}))^{0.5} & [g] \\ m_{TEL\,C} &= 1200 \cdot (\log(\mathrm{Aut}_{TEL}))^{0.6} & [g] \end{cases}$$

We will use the same nomenclature used in the linear case, except for the target function, which, instead of a scalar product with the respective mass vectors m_1 and m_2 will be represented by the functions $f_1, f_2 : \vec{Aut}_X \in \mathbb{R}^n \to \vec{m}_{sys} \in \mathbb{R}^n$

$$\vec{M}_s = \begin{pmatrix} f_{1,1}(Aut_{X,1}) \\ f_{1,2}(Aut_{X,2}) \\ \dots \\ f_{1,n}(Aut_{X,n}) \end{pmatrix} + \begin{pmatrix} f_{2,1}(Aut_{Y,1}) \\ f_{2,2}(Aut_{Y,2}) \\ \dots \\ f_{2,n}(Aut_{Y,n}) \end{pmatrix} = f_1(\vec{Aut}_X) + f_2(\vec{Aut}_Y)$$

Again, we decide to *take the maximum*[35] at the end of the process. This is allowed since the target function are assumed to be strictly increasing. Then we return to $\mathbf{M}_s$ in matrix form, and as a function of the matrices X and Y.

$$\mathbf{M_s} = \left[\left(f_1(X\hat{e}_1) \right), \dots, \left(f_1(X\hat{e}_n) \right) \right] + \left[\left(f_2(Y\hat{e}_1) \right), \dots, \left(f_2(Y\hat{e}_n) \right) \right]$$

Again, we study the derivative of a single subsystem mass, assuming the maximum on the jth row is on the ith and pth column of the X and Y matrix respectively

$$\mathrm{d}m_{sys,j} = \frac{\mathrm{d}f_{1,j}}{\mathrm{d}X_{j,i}}\mathrm{d}X_{j,i} + \frac{\mathrm{d}f_{2,j}}{\mathrm{d}Y_{j,p}}\mathrm{d}Y_{j,p}$$

The derivatives of f_1 and f_2 will depend on the function and the point at which they are calculated. However, as system behaviour is still assumed to be linear, we can express $X_{j,i}$ as a function of Y according to

$$X = [\mathbf{S_1}]^{-1}(\mathbf{E} - [\mathbf{S_2}] \cdot Y) \quad \Rightarrow \quad X_{i,j} = \hat{e}_j^T \cdot ([\mathbf{S_1}]^{-1}(\mathbf{E} - [\mathbf{S_2}] \cdot Y) \cdot \hat{e}_i$$

$$X_{j,i} = C - \begin{pmatrix} S_{j,1} & S_{j,2} & \dots & S_{j,n} \end{pmatrix} \cdot \begin{pmatrix} Y_{1,i} \\ Y_{2,i} \\ \dots \\ Y_{n,i} \end{pmatrix} \qquad \text{where } [S] = [\mathbf{S}_1]^{-1}[\mathbf{S}_2] \quad C \in \mathbb{R} = \text{constant}$$

Which allows us to write each subsystem mass derivative as a function of Y

$$\frac{\mathrm{d}}{\mathrm{d}Y} m_{sys,j} = \frac{\mathrm{d}f_{1,j}}{\mathrm{d}X_{j,i}} \frac{\mathrm{d}X_{j,i}}{\mathrm{d}Y} + \frac{\mathrm{d}f_{2,j}}{\mathrm{d}Y_{j,p}} \frac{\mathrm{d}Y_{j,p}}{\mathrm{d}Y} =$$

$$-\frac{\mathrm{d}f_{1,j}}{\mathrm{d}X_{j,i}} \cdot \begin{array}{c} \begin{array}{ccccc} 1 & \dots & i & \dots & s \end{array} \\ \begin{bmatrix} 0 & \dots & S_{j,1} & \dots & 0 \\ 0 & \dots & S_{j,2} & \dots & 0 \\ & & \dots & & \\ 0 & \dots & S_{j,n} & \dots & 0 \end{bmatrix} \end{array} + \frac{\mathrm{d}f_{2,j}}{\mathrm{d}Y_{j,p}} \cdot \begin{array}{cc} \begin{array}{ccccc} 1 & \dots & p & \dots & s \end{array} & \\ \begin{bmatrix} 0 & \dots & 0 & \dots & 0 \\ & & \dots & & \\ 0 & \dots & 1 & \dots & 0 \\ & & \dots & & \\ 0 & \dots & 0 & \dots & 0 \end{bmatrix} & \begin{array}{c} 1 \\ \dots \\ j \\ \dots \\ n \end{array} \end{array}$$

If we do this for every subsystem, we will obtain the general version of (matrix above for linear case), which we would use in the same way to optimize out target function.

[35]Which is the operation that transforms X into $\vec{Aut}_X$

Algorithm Results

Two comparisons have been performed; the first, using linear mass functions, compares the standard *monolithic* architecture with the flexible one. As mentioned before, in this case the simply redundant one is indistinguishable, as far as mass is concern, from the monolithic one. The second simulation uses the sub linear mass function and confronts the simply redundant system with the flexible one. In this case, the monolithic architecture is not used. This is because the sublinear mass function puts an unfair premium on bigger subsystems, that have an authority to mass ratio considerably high. Even though we established mass as our principle criterion, reliability is also an issue and we can not compare the two architectures.

We also have to chose a metric to asses the effectiveness of these approach. We will report two options

1. Confront only minimum mass system
   As we are interested in only one system design, we may only care about the best option. However, mass optimization may not be the only driving factor. Some subsystem may be available only at a greater cost in complexity or some other *non quantifiable* parameter. Furthermore, at this stage, we are still interested in understanding the overall performances of the flexibility approach, hence the second option.

2. Confront average using each method.
   The optimization using flexibility will be an added cost. Therefore, we might want to have an indication of the expected results to decide whether it is worth it or not. Having an average over a broader sample pool will give us a more representative indication of the expected gains.

Using the linear mass function, we have obtained a **32 % gain** on system mass in the flexible design approach compared with the classical one. On average however, the flexible design is able to decrease system mass of about **8 %**.

Using the sublinear mass function instead we have a **28 % gain** on best system mass but an average **15 % decrease**.

These results are somewhat puzzling, but can be explained. As it turns out, there are subsystem that are overall better than other. While implementing the flexible approach, the algorithm chooses only pair of *different* subsystems. For example, if the primary reaction wheels are the A type, the secondary can not be A as well. Therefore, if a component is significantly better than the other, a design which is allowed to choose it twice has a significant advantage. This possibility was mentioned in chapter 3.
To test this hypothesis, we allow for the flexible algorithm to choose the same component twice. In both cases, we obtain lower system mass even in the smallest mass case. Notably however, average reduction decreases, suggesting that, by adding the same component system, we have a less effective method, which is what we would expect.

The results reported above are clearly preliminary and highly dependent on the models we used to describe each component. At this stage then, they can be seen as an indication of a real possibility of significant improvement, but further development is needed.

Contents

List of Figures