Politecnico di Torino

DIPARTIMENTO DI INGEGNERIA MECCANICA E AEROSPAZIALE
Corso di Laurea Magistrale in Ingegneria Aerospaziale

Master of Science Thesis

Analysis and simulation of instabilities in axial compressors: engine model integration

Candidate:
Luca Tucci

Thesis Advisor:
Prof. Michele Ferlauto

Academic Year 2019-2020

Abstract

The main goal of this thesis is to present a complete simulation of a turbojet engine (namely a GE-J85) which is able to identify anomalies in the compressor, like rotating stall or surge. The simulation introduces the Moore-Greitzer model to the code previously developed by eng. Valerio Timo in his MSc thesis, and aims at recognising the parameters that should reveal the occurrence of one or both of the instabilities.

The actual compressor map was replaced with a new, approximated one, obtained by first nondimensionalising the $\beta_c - \dot{m}_{corr}$ plane, and then by interpolating the experimental values with a polynomial of third order. This allowed to switch from a discrete distribution of points to an analytical function, so that even the part of the map which is behind the surge line, and that gets usually neglected, could be represented.

The Moore and Greitzer's differential equations were then introduced in the engine model, with the aim to perform a transient simulation that could - under specific circumstances - foresee the presence of instabilities. A wide range of values for both the engine and the mathematical model was employed in the analysis to test accuracy and reliability of the simulation. A Greitzer's integration method has been used in a Model Based Predictive Control, with the aim to moderate transient performance when instability is expected. Finally, the results were discussed and eventual improvements to the system were commented.

Contents

List of Figures

List of Symbols

Symbol	Definition
a	Reciprocal time-lag parameter of blade passage
a_s	Speed of sound
A_d	Compressor duct area
B	Greitzer's stall parameter
β	Beta lines value
β_c	Compressor pressure ratio
c	Absolute velocity entering compressor rotor
η	Efficiency
$\dot{m}$	Mass Flow Rate
h°	Total enthalpy
H	Semi-height of compressor characteristic
I_P	Polar moment of inertia
J	Squared rotating stall amplitude
k	Proportional gain
λ	Eigenvalue
L_c	Length of compressor and ducts
l_c	Total aerodynamic length of compressor and ducts (L_c/r_m)
m	Compressor-Duct flow parameter
N	Rotor rotational speed (rpm)
γ	Specific heat ratio
γ_T	Throttle gain coefficient
p	Pressure
p°	Total pressure
p_{std}	Standard pressure at see level
P	Power
ϕ	Flow coefficient
Φ_T	Flow coefficient of throttle duct
ψ	Loading coefficient
Ψ_c	Steady state compressor characteristic
Ψ_{c0}	Compressor characteristic interception with y axis
PSM	Predicted Safety Margin
Q	Blockage coefficient
r_h, r_m, r_t	Hub, mean and tip compressor radii
R	Degree of Reaction
ϱ	Density
SM	Surge Margin
SM_0	Minimum Control Margin

t	Time
T	Temperature
T°	Total temperature
T_{std}	Standard temperature at see level
U	Tangential velocity at mean radius
W	Semi-width of compressor characteristic
ξ	Nondimensional time parameter

Chapter 1

Introduction

The field of jet engines has reached such a level of advancement and technological refinement that contributing to the current state of the art is getting more and more difficult over the years.

Despite this engineering saturation, engine reliability remains under strict scrutiny by every manufacturer. One of the main challenges in keeping such a complex system safe is developing real time simulation algorithms that are able to predict all possible malfunctions, either to avoid them or to stop them in time.

This thesis aims at integrating into a simple engine simulation model one of the most common and accredited mathematical description of compressor instabilities, namely Moore and Greitzer's.

As groundwork we adopted the previous work carried out by V. Timo, who developed the original code used to simulate a basic turbojet operation. This engine configuration was deeply modified in order to be compatible with our instability insurgence model, which was then integrated and exploited for the final results.

All the simulations were conducted in the MATLAB® environment, which provided also the engine for every graphic visualisation. The experimental data necessary to conduct the simulations were obtained from public archives (especially by NASA publications and technical memorandums) and from commercial software *Gas Turbine Simulation Program*.

The ultimate goal of this work was to induce and then recognise unstable behaviours during engine operation, setting the basis for a possible future use in advanced engine control systems. The choice of a rather basic engine model, such as the General Electric J85, eliminates the difficulties determined by more complex architecture: the ability to be extended makes this model versatile and predisposed for a wide range of applications.

1.1 Modelling and control of compressor instabilities

The literature on compressors is broad ad various. A general outline on the subject can be found in articles by Ferguson (1963) or Cohen *et al.* (1996), or in many technical and academic books.

The study of non-stationary phenomena in aeronautical compressors has been conducted ever since the theory of system controls started being exploited in engine design. Many authors analysed all possible types of instabilities, ranging from combustor, to aeroelastic, to aerodynamic. This last category is what we are ultimately interested in. Surge and rotating

stall are the two most common examples of aerodynamic instability, and are thoroughly described by Moore (1981), Jager (1995) and Greitzer (1976).

Of particular relevance is the study conducted by Greitzer in 1976. In the first part of his work he analysed the transient response of a compression system to external perturbations, developing an analytical model that described the effect of those perturbations based on the value of a critical parameter. In the second part, Greitzer applied these results to a set of experimental cases, comparing the numerical predictions to actual stalls and surges situations. In those same years Moore conducted researches on compressor instabilities, publishing extensive studies (April 1984) on the effects of limit cycles and small disturbances on transient operation. In 1986 Moore and Greitzer published a paper that brought together their experience in the field, creating a mathematical model that used a set of three partial differential equations to describe the time evolution of a rotating-stall cell during a transient. Moore and Greitzer model has been generally regarded with great interest in the compressor instability field, and has been used and modified many times in the following years. Its peculiarity is that unstable behaviours are included directly in the mathematical formulation of the problem, and are no longer treated as separate anomalies.

Multiple authors have applied the method to different studies. Shahriyari *et al.* (2019) developed a instability model for low speed contra-rotating Fans; Gravdhal (1998) extended the original model with a close coupled valve to derive a non-linear controller; Nieuwenhuizen (2008) analysed Greitzer's parameters through a comparison with Van der Pool equation; Behnken (1997) included air injection effect in Moore and Greitzer model as a tool for control development.

1.2 Contribution of the present work

The present thesis extends a basic engine simulation environment to make it compatible with the Moore and Greitzer model. The original simulation performed both steady-state and transient analysis of a J85 single spool turbojet. Compressor and turbine maps were imported as discreet experimental data matrices, so intermediate values were computed by means of linear interpolation.

The first part of our work consisted in developing an interpolation algorithm that could shrink the whole set of compressor data into an analytical function, so that a predictive numerical model could replace empirical values every time a point on the map was needed. This was achieved through a least square method that approximated the data with the third order polynomial proposed by Moore and Greitzer. As soon as the approximation was satisfactory, the new numerical compressor model was introduced in the simulation, which was now independent from the fixed, experimental values, and could evaluate freely any point on the map.

The core section of the thesis consisted in the actual integration of the stall model in the transient analysis. In a Model Based Predictive Control logic, the equations that regulated the engine operation were coupled with Moore and Greitzer's third order system, which was used as a means to foresee instability insurgence. For every time step of the engine simulation, the current system variables were integrated through Greitzer's model, to evaluate whether the engine was moving towards stall or surge. A limit threshold on the flow coefficient was set, and a Predicted Safety Margin was computed to discern between safe and unsafe operation points. This parameter was then used in a proportional control over the fuel flow injection, which would be reduced if the full value would bring the engine towards instability.

This system was finally employed in various transient performance simulations, showing decent efficiency in predicting unsafe behaviours and regulating engine operation accordingly.

1.3 Outline of this thesis

The thesis will develop as follows:

Chapter 2: A comprehensive theoretical introduction is given. Compressors are described in terms of performance and design parameters, and aerodynamic instabilities are outlined.

Chapter 3: The Moore and Greitzer model is presented, including the constitutive equations and the defining parameters. A backstepping integration method is introduced to solve dynamic system, and numerical examples of rotating stalls and surges are commented.

Chapter 4: Compressor maps features are introduced. Interpolation requirements are discussed, with particular stress on the main issues. The chosen interpolation technique is presented, and the approximation results are shown. Alternative methods and their limits are then listed.

Chapter 5: The simulation environment is presented. With the help of flowcharts, steady-state and transient analysis are described, and the new approximated compressor map is integrated in the model. After selecting a specific fuel schedule, stationary and transient working lines are derived and represented on the compressor map.

Chapter 6: The problem of integrating the Moore-Greitzer model in the simulation is tackled. A stability analysis is conducted to show the unstable regions of the compressor map, and an example of proportional control in surge avoidance is described. A Model Based Predictive Control logic is then proposed through Greitzer's method to foresee and control instabilities during operation. Transient performance with predictive control is analysed through various case studies.

Chapter 2

Theoretical background

2.1 Axial compressors

Compressors are used in a wide range of engineering applications, either as a single tool or inside a more complex device. In aviation, all kind of jet engines need to compress the air coming from the inlet in order to produce a denser and hotter flow, so compressors represent a key component of the whole structure.

Two different types of architectures are usually employed:

- **centrifugal** compressors use an *impeller*, a rotating disk placed in a specific housing, to direct the incoming flow towards its rim and therefore accelerating it; the following *diffuser* is then appointed to convert the resulting kinetic energy into pressure by means of a divergent duct.

- **axial** compressors are composed by rotating discs (*rotors*) alternated to static discs (*stators*) put in sequence, upon which arrays of airfoils progressively compress the airflow.

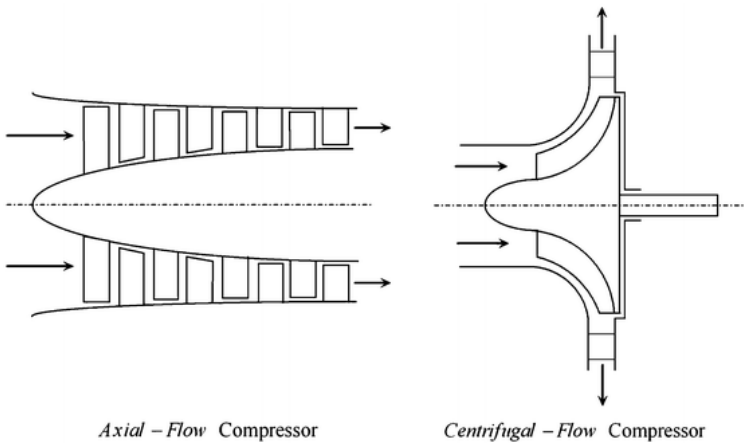


Figure 2.1: *Axial and centrifugal compressor comparison*

In this thesis we will focus specifically on axial compressors, as they are most commonly used in basic turbojets configurations.

2.1.1 Performance and project parameters

Axial compressors are normally represented as a convergent duct. Assuming that the axial flow speed remains almost constant and that its density increases throughout the axis of the machine, in order to satisfy the mass balance the cross section area needs to get smaller:

$$\dot{m} = \varrho A c_x = \text{const} \qquad \Longrightarrow \qquad \varrho \uparrow, \quad A \downarrow .$$

The performance of a compressor is determined by its ability to convert the speed of the incoming air into pressure. That is achieved by deflecting the flow direction in such a way that it gradually slows down and gains enthalpy while passing through airfoil-like blades.

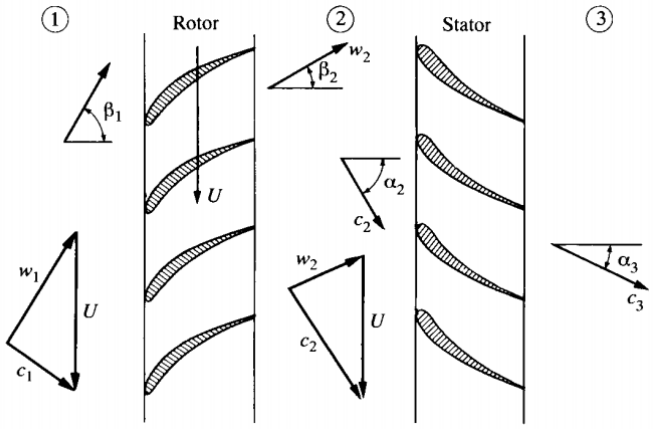


Figure 2.2: *Section of a compressor stage and velocity triangles*

Figure 2.2 shows a typical section of an axial compressor stage at mean radius. As previously mentioned, a stage is made of two consecutive parts:

1. the **rotor** is the moving part of the stage. It receives the air from either the inlet or a previous stage and, by combining the relative velocity w (to the blade geometry) with the rotating speed of the disc U, accelerates it to produce kinetic energy. At the same time, the relative velocity between the blades decreases to pressurize the flow.

2. the **stator** is the motionless part of the stage, as it is coupled with the external housing of the engine. Its purpose is to further increase the air pressure by transforming the kinetic energy obtained in the rotor. At the stator exit the absolute velocity is usually smaller than at the entry ($c_3 < c_2$). A common hypothesis is that the flow direction leaving the stator is the same of the flow entering the rotor (*repetitive stage*), so usually $c_1 = c_3$.

One way to express the ability to extract compression energy from the rotor motion is by relating the power of the machine to the enthalpy jump between the exit and the entry. The following equation is derived both from the energy balance in turbomachinery and from

conservation of angular momentum:

$$L_{in} = \frac{P}{\dot{m}} = h_3^\circ - h_1^\circ.$$

Once we define the stage efficiency as

$$\eta_{st} = \frac{h_{3,is}^\circ - h_1^\circ}{h_3^\circ - h_1^\circ}$$

we can then determine the pressure rise for each stage:

$$\frac{p_3^\circ}{p_1^\circ} = \left(1 + \eta_{st}\frac{\Delta T^\circ}{T_1^\circ}\right)^{\frac{\gamma}{\gamma-1}}. \tag{2.1}$$

It is common practice, when describing the general performance of a compressor (either axial or centrifugal), to introduce some new variables, usually referred to as **project parameters**. There are 3 of these, and their purpose is to help choosing the compressor design point during the project phase.

Table 2.1.1 lists the project parameters and clarifies their function.

Name	*Expression*	*Definition*
Degree of Reaction	$R = \frac{h_2 - h_1}{h_3^\circ - h_1^\circ}$	Distribution of work between rotor and stator
Loading Coefficient	$\psi = \frac{L_c}{U^2}$	Amount of work produced in a single stage
Flow Coefficient	$\phi = \frac{c_x}{U}$	Incidence of mass flow in the axial direction

The figure below shows how the project parameters are related, reporting also their dependence to the blade inlet angles α_1 and β_2.

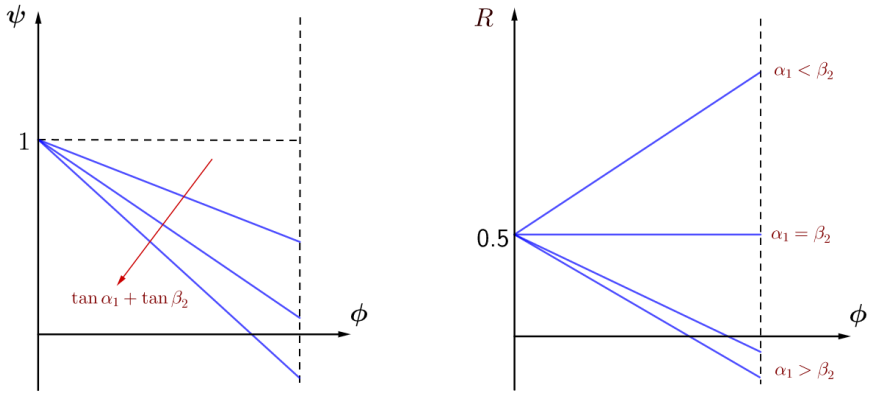


Figure 2.3: *Relations between project parameters*

2.1.2 Compressor map

So far we discussed the characteristics of a compressor when it is considered an independent device, extracted from a more complex architecture.

When introducing the compressor into an engine model, however, it is necessary to find a way to associate the functioning of the single part to how the whole system works. For this reason **maps** were developed that could show the relation between what enters the compressor, namely the amount airflow $\dot{m}$, and the compression process result, that is the pressure ratio β_c.

Compressor maps are usually obtained by means of *rig tests*, and therefore are based experimental data. Successive chapters will show how to go about a possible approximation of such data, thus obtaining an analytical expression.

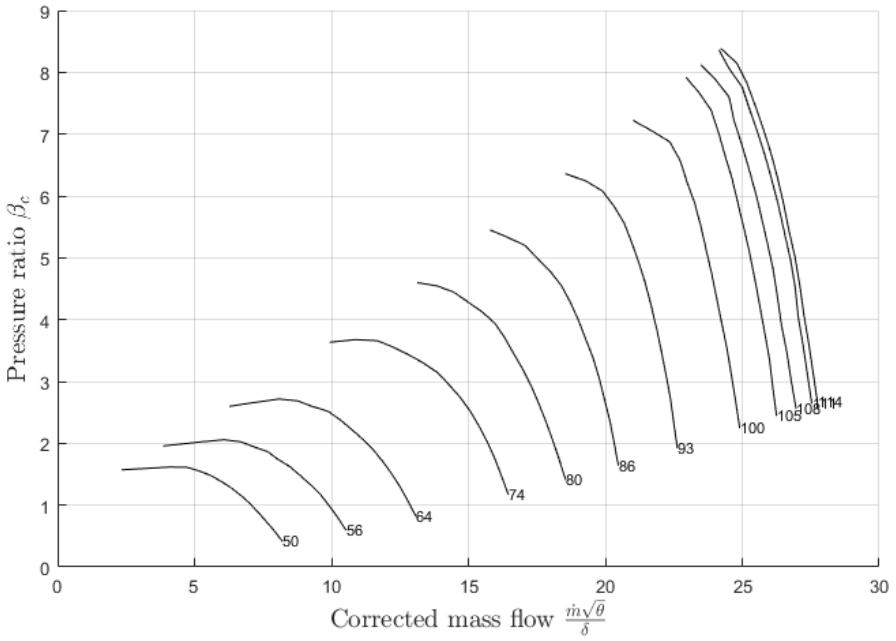


Figure 2.4: *Example of compressor map*

Figure 2.4 shows an example of a compressor map. In order to make the involved parameters independent from the atmospheric conditions, these are normally reported in a **corrected form**. This means that the terms are divided by standard temperature and pressure values, as follows:

$$\begin{aligned} \dot{m} \quad &\Longrightarrow \quad \frac{\dot{m}\sqrt{T^\circ/T_{std}}}{p^\circ/p_{std}} \quad \Longrightarrow \quad \frac{\dot{m}\sqrt{\vartheta}}{\delta} \\ N \quad &\Longrightarrow \quad \frac{N}{\sqrt{T^\circ/T_{std}}} \quad \Longrightarrow \quad \frac{N}{\sqrt{\vartheta}}. \end{aligned}$$

The figure also shows that pressure ratio and corrected mass flow vary as a function of the corrected rpm. The influence of this further parameter results in a family of curves which spreads towards the upper right corner of the plot: for every value of corrected speed, we have a different $\dot{m}_{corr} - \beta_c$ curve.

Briefly, we can summarise that:

1. as the mass flow **increases** with constant speed, the pressure ratio **decreases**;

2. as the corrected speed **increases** with constant mass flow, the pressure ratio **increases**.

2.2 Compressor instabilities

All compressor systems are subjected to many kind of unstable behaviours, spanning from combustion irregularities to aeroelastic flutter. The type of instabilities this thesis will focus on is specifically the **aerodynamic** one.

Aerodynamic instabilities can present themselves in two main phenomena, **rotating stall** and **surge**, and can deeply affect the performances of the turbomachine, leading in extreme cases to complete failure.

We will now describe in detail both these dangerous malfunctions.

2.2.1 Rotating stall

Rotating stall is a domino sequence which happens when one or more rotor blades are heavily loaded, for instance as a consequence of **nonuniform flow distribution**.

When a blade, or even just a region of it, is subjected to an instantaneous peak in load, the associated lift coefficient tends to move beyond its tolerable limit, so stall occurs. The particularity here is that once the instability is triggered, it is not contained into that same area of the disc, but the aerodynamics lead it to spread cyclically. One or many **stall cells** are then created, and while propagating around the compressor annulus they determine a collapse in both the compression efficiency and the amount of mass flow passing thorough it.

Let's now try to describe the physical process that results in rotating stall (for a more thorough explanation refer to Emmons *et al.* (1955)).

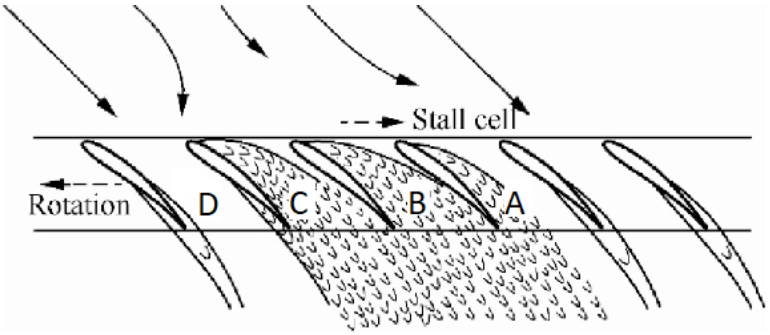


Figure 2.5: *Rotating stall in a blade array*

Figure 2.5 shows a typical compressor blade cascade operating at high angle of attack. Let's now imagine that the flow incidence investing blade B is so high that it stalls: the flow separation on its back reduces the passage area so that the flow cannot completely go through. What happens now is that this partial blockage deflects the mass flow in two different directions:

- part of it is oriented towards blade A, dangerously increasing the incidence of the flow which was already moving towards it;

- part of it is oriented towards blade C, this time reducing the incidence of the oncoming air and bringing it back to a more tolerable level.

The result of this split-up is that the stall moves from blade B to blade A, whilst blade C operates in a stable condition. The phenomenon is now repetitive because after A is stalled, blade B will receive the low-incidence flow it deflects, starting again to work in nominal conditions. Basically then, the stall cell as a whole tends to move rightwards with a **propagation speed** u_s, whereas the blades move leftwards with the tangential rotation speed U. According to the most accredited literature (see Greitzer (1980)), the ratio between these two velocities can vary in the 20-70% range.

Insurgence of rotating stall isn't usually a fatal problem in the compressor functioning: when it is detected, control systems intervene to restore the nominal operation. Nonetheless, there can be different situations where such an instability can become very dangerous for the whole turbomachine. As previously mentioned, the first drawback (and also the easiest to ascertain) is in performance: the compressor cannot produce the pressure increase that is nominally required due to inactive areas of the annulus.

Another significant consequence of a stalled compressor may be **vibrations**. When the blades cyclically pass in and out the stalled annulus area, they are mechanically solicited by the induced vibrations. If those move with a frequency that coincides with one of the acoustic modes in the inlet or discharge lines, the blades may even incur **resonance**, with the possible risk of **fatigue failure**.

Risky behaviours can arise also when trying to address the problem of stall by using the throttle. Rotating stall is known to be a source of **hysteresis** in the compressor functioning, as thoroughly described by many authors. Without going in too much detail (the issue will be tackled later), what happens can be explained with the help of the following image.

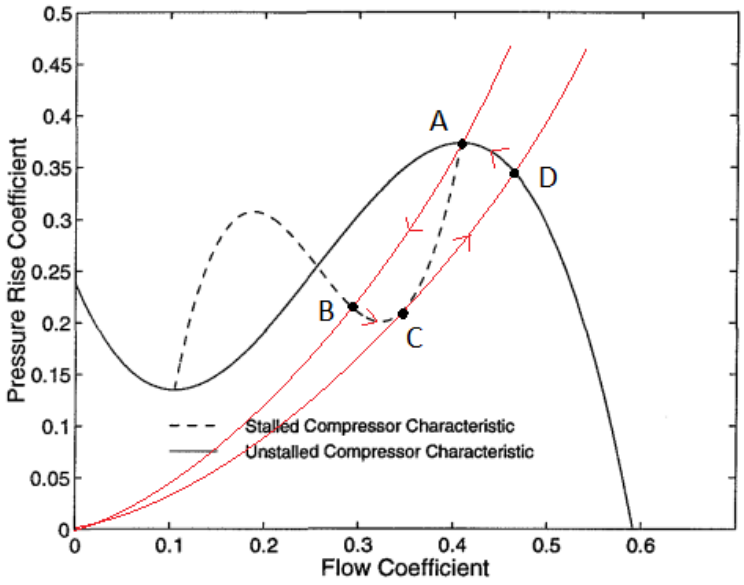


Figure 2.6: *Hysteresis cycle due to rotating stall*

Figure 2.6 show the hysteresis cycle caused by a rotating stall on a compressor map. The solid line represents the usual map with constant velocity, whereas the dashed line refers to the characteristic of a stalled compressor. The red lines, finally, are two different throttle characteristics.

Let's first consider point A, where the turbomachine is set to work in a safe region, and therefore no stall is detected. If any perturbation produces a change in the system, the working point moves along the throttle characteristic towards a new equilibrium, which this time is on the stalled characteristic (B). To bring the compressor back on a stable point the throttle is closed, so the new working point moves along the dashed line towards C, which is on a less sloped throttle characteristic. Now the system can go back to the stable region of the map (point D), and by reopening the throttle again on the initial point, where the cycle can repeat itself.

The severity of this malfunction can be measured by observing the area contained within $A-B-C-D$: if the slopes of both compressor and throttle characteristics are sufficiently high, the hysteresis cycle becomes less dangerous.

2.2.2 Surge

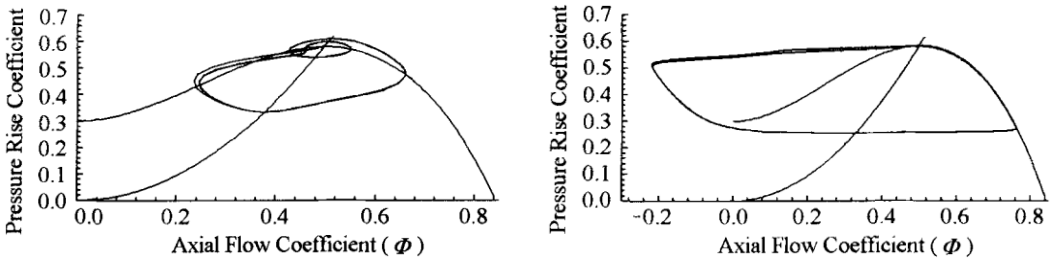
Surge is a form of instability which involves the whole environment where the compressor is placed (*global instability*). It presents as the airflow oscillation in the axial direction of the machine, resulting in a limit cycle in the compressor characteristic.

By measuring the entity of this instability (or more specifically its oscillations amplitude), we can usually distinguish between two types of surge:

1. **mild surge**, where the fluctuations affect only the mass flow rate going through the turbomachine, but no flow inversion is determined;

2. **deep surge**, where mass flow can even become negative, so reverse flow is produced; fluctuations are much more intense.

Deep surge is clearly the most dangerous of the two, in that the oscillation frequency is so high that the structural integrity of the compressor might get compromised. Two examples of mild and deep surge are displayed below.



(a) *Mild surge* **(b)** *Deep surge*

Figure 2.7: *Limit cycles for the two types of surge*

It is now necessary to clarify some basic notions on **stability**, which will be also useful later on in this thesis. When considering a system described by a curve, the simplest way to discern stable and unstable regions is by observing the slope:

- when the curve slope is **negative** $\left(\frac{\partial\psi}{\partial\phi} < 0\right)$, the system is **stable**, so it is able to naturally absorb a perturbation;
- when the curve slope is **positive** $\left(\frac{\partial\psi}{\partial\phi} > 0\right)$, the system is **unstable**, so a perturbation is indefinitely amplified.

This is why compressor maps usually report the so called **surge line**, which is the explicit limit that separates stable compressor functioning (at its *right*) from the unstable region (at its *left*). Here follows the integration of figure 2.4 with the surge line.

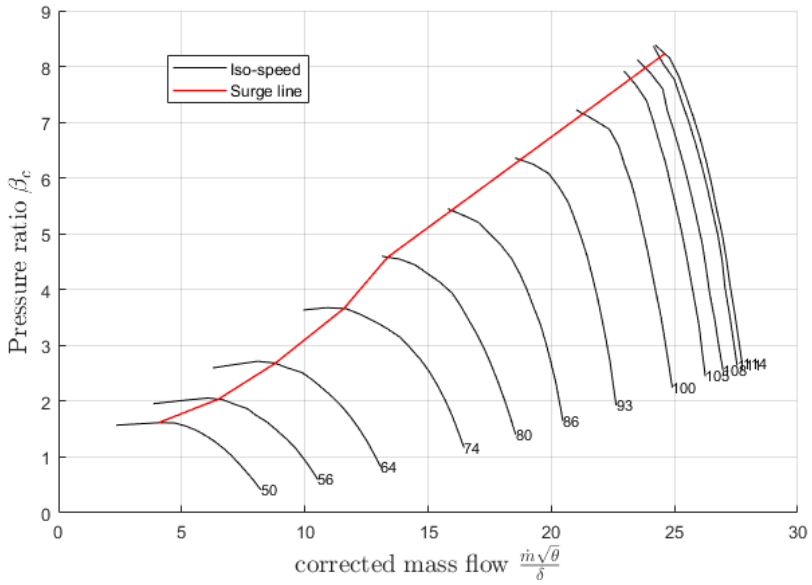


Figure 2.8: *Compressor map with surge line*

Let's now consider a simplified example of a surged compressor characteristic to better understand how the instability can occur.

Figure 2.9 shows a complete deep surge cycle, extended also in the negative half-plane of the mass flow rate. The solid red characteristic belongs to the compressor, while the black dotted one refers to the external circuit, which usually corresponds to the engine system. Point A, where the two intersects, is the design point of the system: it was deliberately taken on the increasing branch of the curve, and so it is prone to instability.

Let's imagine that an abrupt perturbation of the environment brings the pressure ratio to a lower value than the nominal. Then the compressor working point changes to A', to which point A'' corresponds on the external circuit line. The compressor can now process a mass flow rate which is **inferior** to the amount required by the engine ($\phi_{A'} < \phi_{A''}$): after the residual air contained in the circuit runs out, pressure starts to decrease again, up until it reaches the level of points C and E respectively. At this point, as the air demand of the engine is yet not met by the compressor, the machine jumps directly to point D, which is the only one allowing a further depressurisation (we need a $\psi < \psi_C$). In this new configuration (now back on the stable branch of the map) $\phi_D > \phi_E$, so the machine starts accumulating air and the pressure increases: the compressor working point moves up to G.

The compressor now absorbs **more** airflow than the system can process, so the only way this can happen is by means of a pressure rise: the working point necessarily jumps to H. Now the compressor works with an even *negative* mass flow rate, so the capacity rapidly empties and the pressure goes down. As the compressor approaches point B the cycle tends to repeat itself, because $\psi_B = \psi_A$, and the system is back to the initial configuration.

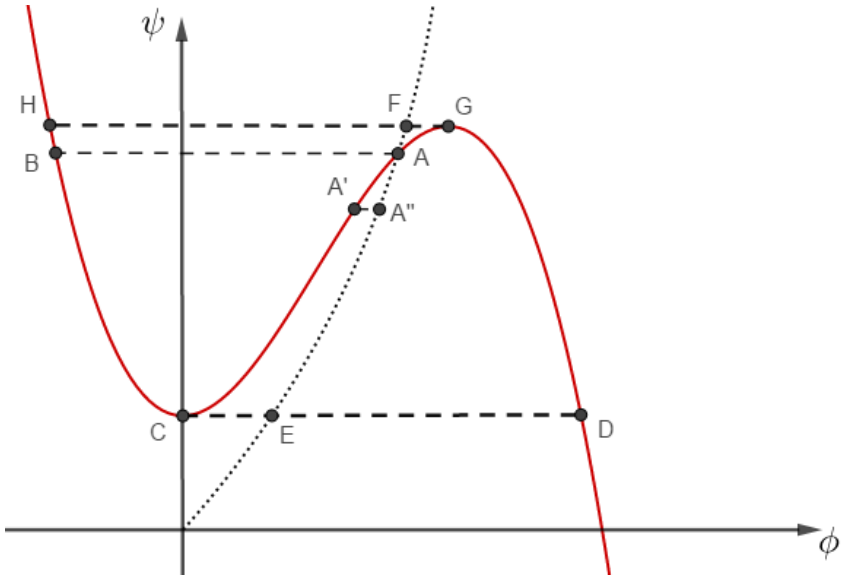


Figure 2.9: *Compressor surge cycle outline*

The complete cycle is then designated by the succession $G \to H \to C \to D \to G$.

Chapter 3

The Moore-Greitzer model

In our theoretical introduction we have outlined how an axial compressor nominally works and what its main forms of instability are. In this chapter we are going to focus our attention on combining these two pieces of information through a model, the one developed by Moore and Greitzer.

3.1 Mathematical formulation

Over the last fifty years, unstable behaviours in compressor systems have been the subject to a variety of academic studies, which even grew in number the moment when predicting all sorts of anomalies in the engine has become so paramount.

Amongst all the different approaches to this problem, the model of **Moore and Greitzer** stands out in terms of accuracy and versatility. It is the reinterpretation of the work previously conducted by eng. E. M. Greitzer in [1], and then integrated by eng. F. K. Moore in 1986 ([2]). What makes the Moore-Greitzer model particularly appealing to the research is its ability to include the rotating stall amplitude as an *intrinsic state* in the dynamic behaviour of the compressor, making it possible to automatically detect it as soon as some parameters exceed their safe boundaries.

The mathematical formulation of the model then consists in a set of PDEs which describe the time evolution of the compressor system: a bifurcation analysis of the solutions will then provide the stability information we look for.

Figure 3.1 outlines the architecture the theory is built on.

The system consists of a basic compressor-plenum-throttle chain where the following hypothesis are met:

1. the plenum dimensions are bigger than those of any other component (either compressor or ducts), so all its physical dimensions can be considered **static** (accelerations are negligible): the plenum pressure is *uniform* but not necessarily stationary;

2. the airflow is **incompressible** in the compressor but **compressible** in the plenum as it acts like a gas spring, so there is a delay between the throttle opening and the consequent pressure rise;

3. the flow is two-dimensional, so the compressor hub to tip ratio is sufficiently high;

4. the throttle can be represented by a valve located after the plenum.

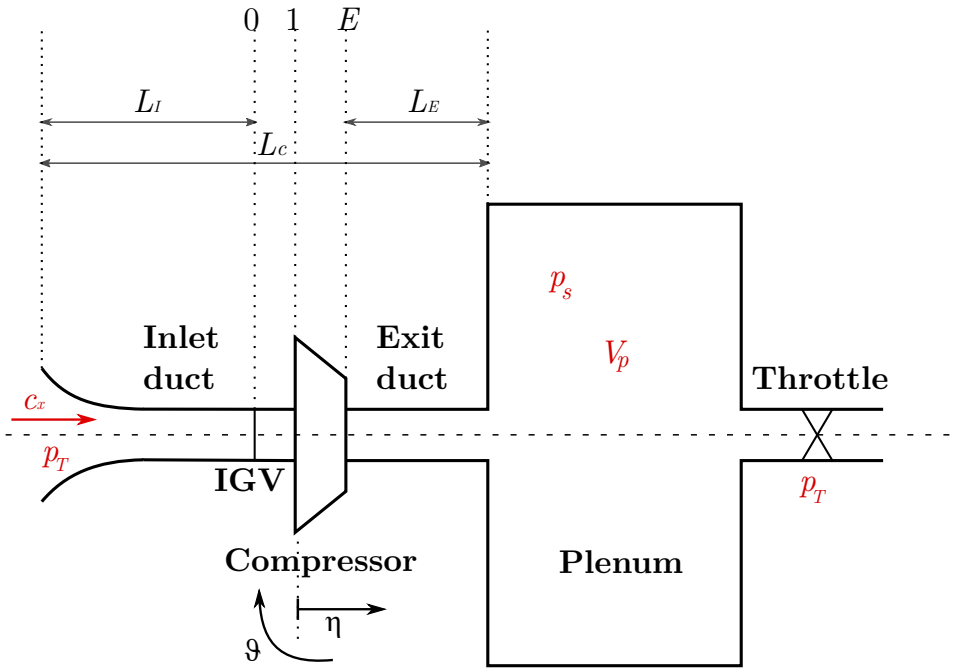


Figure 3.1: *Moore and Greitzer model configuration*

Following the procedures thoroughly explored in [2], we can get to the general expression of the Moore-Greitzer model equations:

$$\begin{cases} \Psi(\xi) = \Psi_c(\Phi - Y_{\vartheta\vartheta}) - l_c \dfrac{d\Phi(\xi)}{d\xi} - mY_\xi + \dfrac{1}{2a}(2Y_{\xi\vartheta\vartheta} + Y_{\vartheta\vartheta\vartheta}) \\ \dfrac{\partial\Phi(\xi)}{\partial\xi} = \dfrac{1}{l_c}\left[\dfrac{1}{2\pi}\displaystyle\int_0^{2\pi} \Psi_c(\Phi - Y_{\vartheta\vartheta})d\vartheta - \Psi(\xi)\right] \\ \dfrac{\partial\Psi(\xi)}{\partial\xi} = \dfrac{1}{4l_cB^2}\left(\Phi(\xi) - \Phi_T\right) \end{cases} \tag{3.1}$$

Dynamic system 3.1 reports the three balance equations associated to the configuration above, namely the *local momentum* balance, the *annulus averaged* momentum balance and the *plenum mass* balance. All the quantities are used in their non-dimensional form, so more specifically $\eta = x/R$ is the axial coordinate, with zero at compressor entry; $\xi = \frac{Ut}{R}$ is the non-dimensional time, where R is the mean compressor radius; ϕ is the local flow coefficient in the annulus; ϑ is the circumferential position around the annulus.

Based on these coordinates all the main system variables are defined. In particular we highlight the *pressure rise coefficient*

$$\Psi = \frac{p_S - p_T}{\varrho U^2},$$

the *axial flow coefficient*, which corresponds to the one defined in table 2.1.1, but annulus-averaged

$$\Phi(\xi) = \frac{c_x}{U} = \frac{1}{2\pi}\int_0^{2\pi} \phi(\vartheta, \xi)d\vartheta,$$

the **throttle flow coefficient**

$$\Phi_T = \gamma_T \sqrt{\Psi},$$

where γ_T is the *throttle gain*, and finally Y, the *disturbance potential at* $\eta = 0$, such that

$$\int_0^{2\pi} Y(\vartheta, \xi) d\vartheta = 0.$$

A complete understanding of equations 3.1 requires also the definition of a few parameters. Beside m, which is just related to the compressor-duct flow, two important coefficients are:

- the *effective flow passage length* through compressor and ducts (measured in wheel radii)

  $$l_c = l_i + \frac{1}{a} + l_E,$$

  where a is the reciprocal *time lag parameter* of blade passage;

- the **Greitzer parameter**

  $$B = \frac{U}{2a_s}\sqrt{\frac{V_p}{A_d L_c}}$$

  (read [1] for details), where V_p is the plenum volume, a_s is the speed of sound, A_c is the flow area and U is the tangential velocity at mean radius.

Those reported in 3.1 are the Moore-Greitzer equations written in their generic expression, where no specific relation for the compressor characteristic has been established.

Before addressing this further issue, however, it is possible to use this general system to surmise how the model behaves in special cases like pure rotating stall and pure surge, that is if no coupling between the instabilities is observed.

When pure surge is detected the equations becomes much simpler, because the absence of any circumferential variation determines $Y = 0$. As a result the first two equations both reduce to

$$\Psi(\xi) = \Psi_c - l_c \frac{d\Phi(\xi)}{d\xi}.$$

On the other hand, in case of pure rotating stall there is no time evolution in the system, so every derivative with respect to ξ is null: the third equation in 3.1 becomes simply

$$\Phi(\xi) = \Phi_T.$$

Now that this clarification has been made we can try to specify how the function Ψ_c, so far remained generic, can appear. One of the main contributions of the Moore-Greitzer model was to propose a law that could describe the usual characteristics of the compressor map with an easy analytical expression. That law is in the form of the following cubic polynomial, whose geometry is defined some key parameters, as shown in the figure below:

$$\Psi_c = \Psi_{c0} + H\left[1 + \frac{3}{2}\left(\frac{\Phi}{W} - 1\right) - \frac{1}{2}\left(\frac{\Phi}{W} - 1\right)^3\right] \tag{3.2}$$

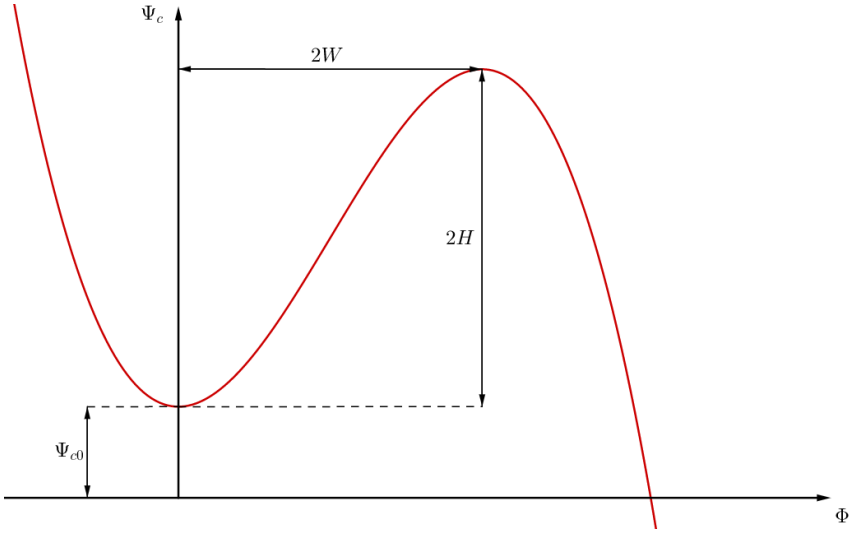


Figure 3.2: *Compressor characteristic in Moore-Greitzer model*

If we introduce 3.2 into 3.1, and then apply a Galerkin procedure to the main variables, we can develop a new model to solve the problem. First we assume that the potential Y can be expressed as a single harmonic with unknown amplitude A and phase r:

$$Y = WA(\xi)\sin(\vartheta - r(\xi)).$$

Then, by defining the new variable $J = A^2(\xi)$ as the **squared rotating stall amplitude**, we finally get to the definitive form of the Moore-Greitzer model.

$$\begin{cases} \dfrac{d\Psi(\xi)}{d\xi} = \dfrac{W/H}{4B^2}\left[\dfrac{\Phi}{W} - \dfrac{\Phi_T}{W}\right]\dfrac{H}{l_c} \\ \dfrac{\partial\Phi(\xi)}{\partial\xi} = \left[-\dfrac{\Psi - \Psi_{c0}}{H} + 1 + \dfrac{3}{2}\left(\dfrac{\Phi}{W} - 1\right)\left(1 - \dfrac{J}{2}\right) - \dfrac{1}{2}\left(\dfrac{\Phi}{W} - 1\right)^3\right]\dfrac{H}{l_c} \\ \dfrac{dJ(\xi)}{d\xi} = J\left[1 - \left(\dfrac{\Phi}{W} - 1\right)^2 - \dfrac{J}{4}\right]\dfrac{3aH}{(1+ma)W} \end{cases} \tag{3.3}$$

3.2 Numerical solution technique

In the last paragraph we described the characteristics of the model developed by Moore and Greitzer, outlining the parameters it is based on and its complete analytical formulation. Starting from that introduction, we are now interested in finding a way to integrate that model, so that concrete instabilities can be simulated.

To do so, a numerical integration method was adopted to deal with the non linear equations composing dynamic system 3.3.

It is clear that solving 3.3 means finding the time evolution of the three main variables Φ, Ψ, J, once the necessary model parameters have been set. In figure 3.3 below is reported the flowchart developed to this purpose.

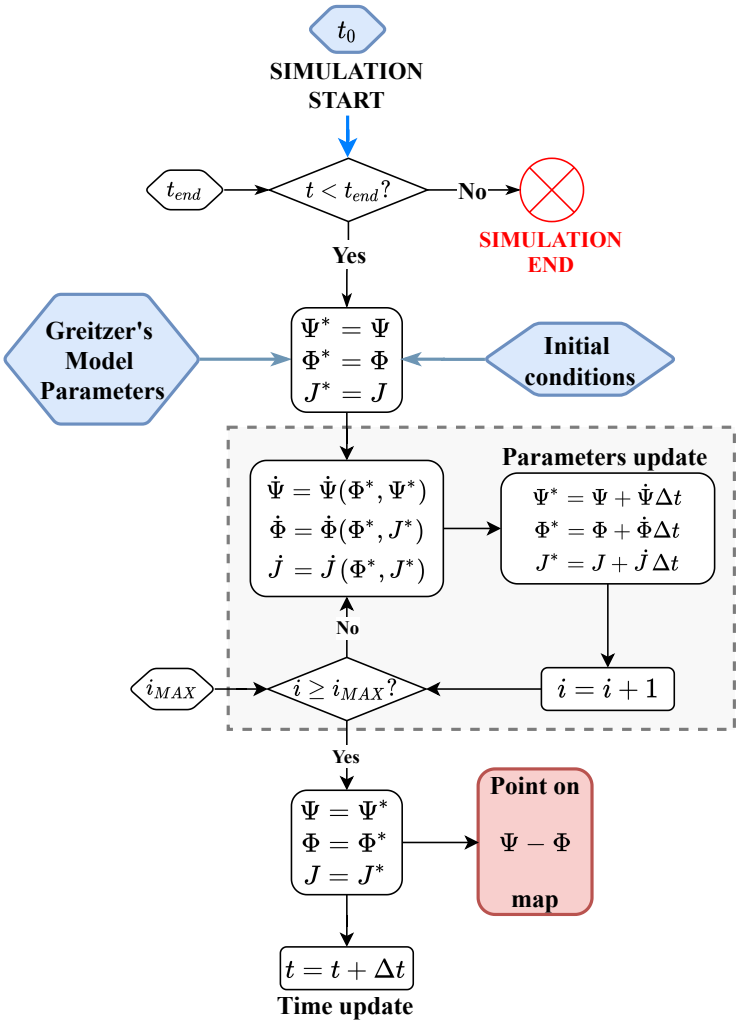


Figure 3.3: *Greitzer model simulation flowchart*

The simulation operates in the timespan between the initial time t_0 and the final time t_{end}, and for all intermediate time-steps computes the values of Ψ, Φ and J. For every t

we can then represent a point on the $\Psi - \Phi$ plane, so that at the end we can visualise the presence of possible instabilities.

A recursion loop, which is performed for a fixed number of times i_{MAX}, is used to to bring the derivatives to convergence, with a technique usually referred to as **backstepping integration**. More specifically, actual variables Φ, Ψ, J are accompanied by support variables Φ^*, Ψ^*, J^*, which are used to advance over the time interval by iteratively computing $\dot{\Phi}, \dot{\Psi}, \dot{J}$ through system 3.3. The peculiarity is that the time integration is only used to update these derivatives, whereas the system variables are modified only once for the last iteration.

Usually three to five iterations are enough to get sufficiently accurate values of the derivatives: when the cycle ends the new variables are given as outputs, and the simulation finally moves to the following time value.

The problem needs to be initially defined with the help of some key inputs, such as, in particular:

- **Greizter's parameter** B,
- duct **lengths** l_c, l_I, l_e,
- stationary **compressor** characteristic **parameters** H, W, Ψ_0,
- throttle **gain** γ_T.

For the first iteration we also need to externally set the values for Ψ, Φ and J, which are usually chosen on the compressor characteristic.

We can briefly summarise how this simulation works as follows:

1. An initial time input is given to start the simulation;
2. The constant parameters of the model are externally set and an initial point on the $\Psi - \Phi$ map is introduced for the first loop;
3. a recursive method is used to compute the dotted variables through the model's equations, and after a fixed number of iterations convergence is assumed;
4. the new derivatives are adopted to update the current system variables, which are then represented on the compressor characteristic;
5. time is increased, and until t_{end} is reached the whole process is repeated.

As a matter of fact, we are replacing the usual, **stationary** compressor characteristic with a new, **dynamic** one, that changes with time according to stall amplitude J value:

$$\Psi_c = \Psi_{c0} + H\left[1 + \frac{3}{2}\left(\frac{\phi}{W} - 1\right)\left(1 - \frac{J}{2}\right) - \frac{1}{2}\left(\frac{\phi}{W} - 1\right)^3\right] \tag{3.4}$$

It is important to point out that this is only one of the many methods that would appropriately solve the Moore and Greitzer model. Results comparison with commercial algorithms can be carried out if need be.

3.3 Examples of instabilities and their features

With the adoption of the numerical model we developed in the previous paragraph we were able to perform some simulations, framing various examples of surges and rotating stalls.

It is important to specify that, among the many parameters involved in the problem definition, the most determinant in the insurgence of one instability rather then the other is Greitzer's B parameter:

- **small** B values bring the system into stall;
- **large** B values create surge conditions.

We are now going to discuss four interesting cases, two for each kind of instability. For all the simulations the following parameters were left constant:

Parameter	*Value*
Ψ_{c0}	1.5
W	0.0492
H	1.5
l_c	2.03
a	0.3
m	1.75
t_0	0
$\dot{\Psi}(0)$	0
$\dot{\Phi}(0)$	0
$\dot{J}(0)$	0
$J(0)$	0.1

3.3.1 Stall examples

The first example of rotating stall we achieved was based on these input parameters:

Parameter	*Value*
ϕ_0	0.1049
ψ_0	4.458
B	0.01
γ_T	0.0497
t_{end}	3

The starting point was picked on the right branch of the characteristic, with a throttle gain derived accordingly through the usual relation $\gamma_T = \frac{\phi_0}{\sqrt{\psi_0}}$. The B value was chosen small enough to induce stall. The results are reported above.

From figure 3.4 we can observe that with this set of values we obtain a pretty small instability, which is circumscribed in a limited region of the map. To get a clearer view of the stall we can refer to figure 3.5, which is a magnification of the latter. The deviation from the stationary characteristic starts from the operating point, where it returns after evolving around the throttle line. Should we zoom further in, we would see the spiral trend we expect from rotating stall develop around the starting point.

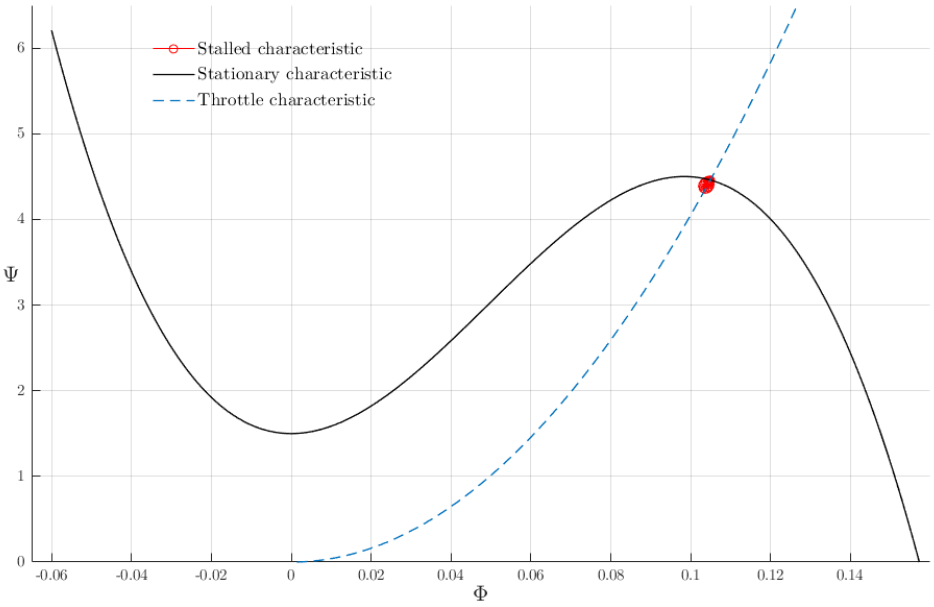


Figure 3.4: *Example of a small rotating stall*

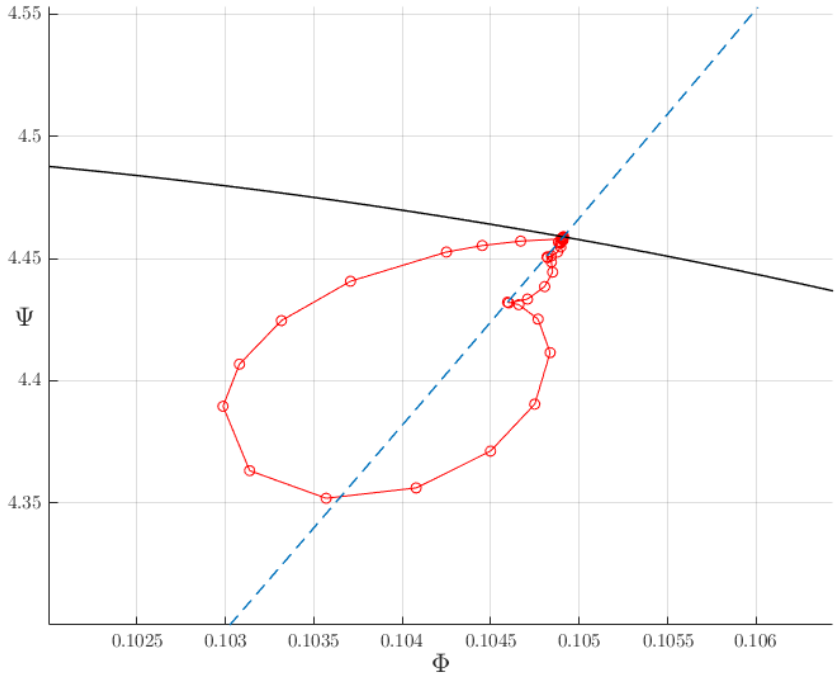


Figure 3.5: *Magnification of the stalled region*

To observe how the results behave we may analyse the $\phi - t$ and $\psi - t$ evolution. In figure 3.6 below we may see that both the coefficients start from their assigned values and, after a brief and damped oscillation, get back to them.

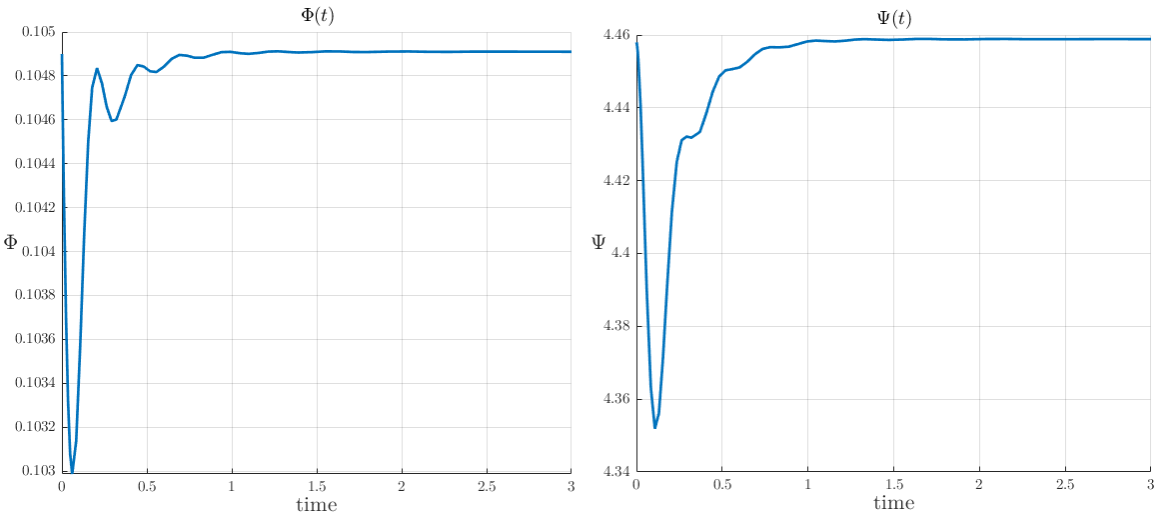


Figure 3.6: *Time evolution of nondimensional variables*

Very important in recognising stall is, finally, parameter J, which needs to be non-zero up until convergence is met:

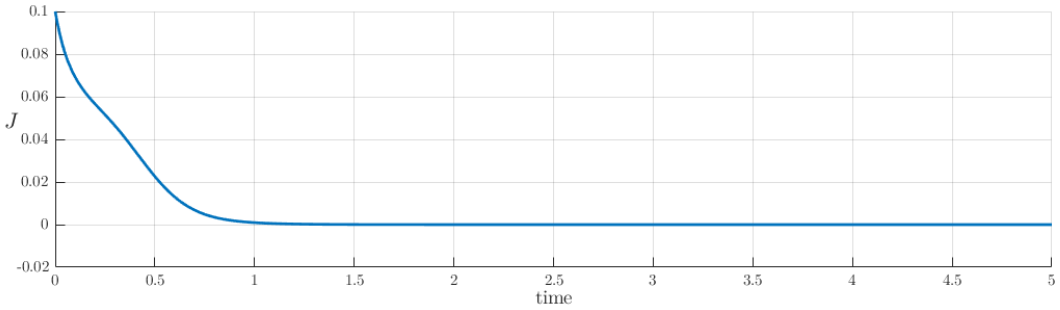


Figure 3.7: *Stall amplitude evolution*

The moment starting point and operating point of the simulation do not coincide anymore, the instability tends to become more evident. For our second example of rotating stall we externally set the throttle gain γ_T, together with the following inputs:

Parameter	*Value*
ϕ_0	0.1218
ψ_0	3.9022
B	0.01
γ_T	0.048
t_{end}	5

What we obtained is represented below.

From the starting point the solution moves towards the throttle law set from the outside, and to do so follows a spiral trend which is much more definite than in the previous case.

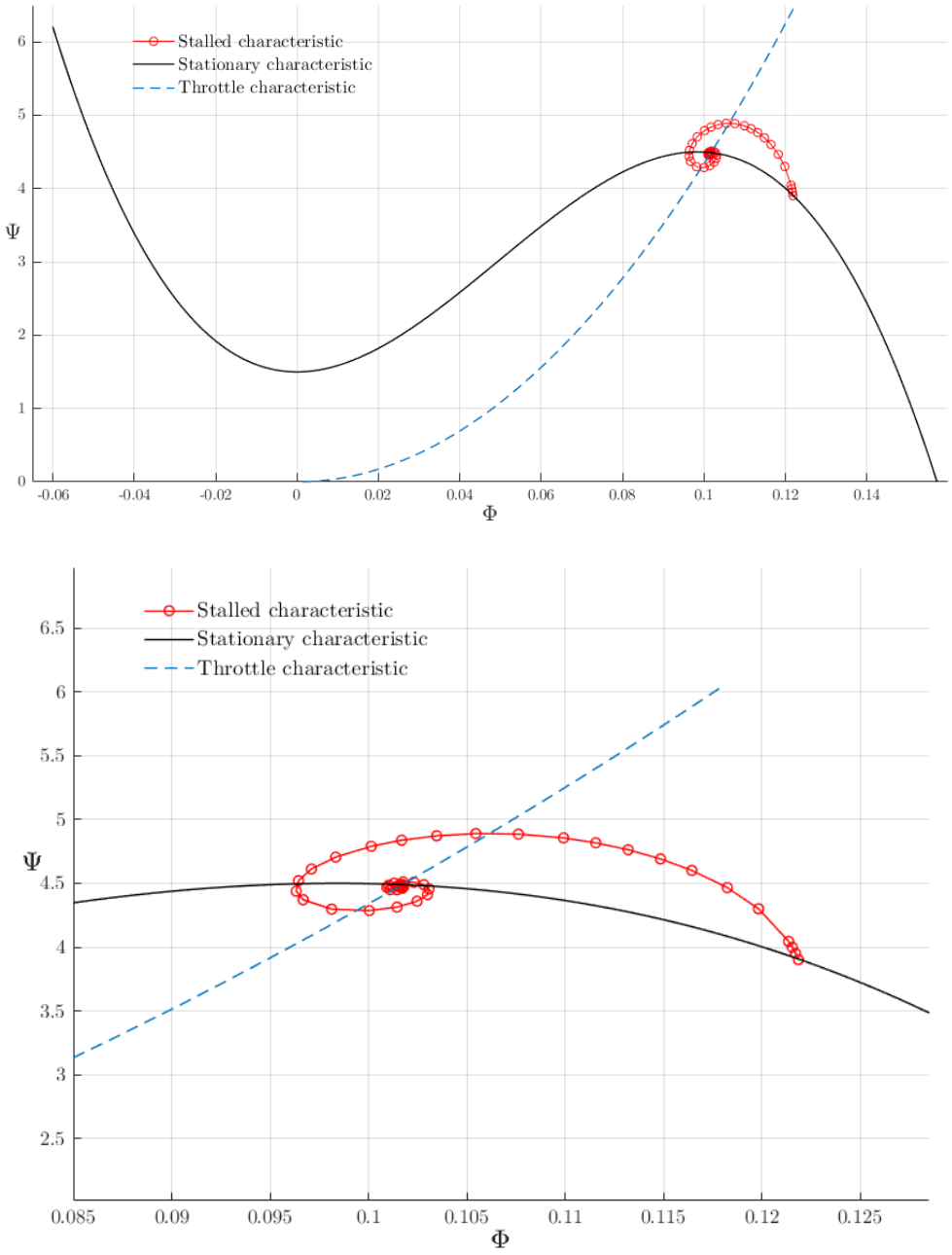


Figure 3.8: *Example of a heavy rotating stall*

The increased severity of this stall appears clearly also from the parameters time evolution. Figure 3.9 shows that the system remains stable, because the variables eventually converge to the operating values, but before settling on the equilibrium wider oscillations arise.

Once again, we can also see the square stall amplitude J changing during the first part of the simulation.

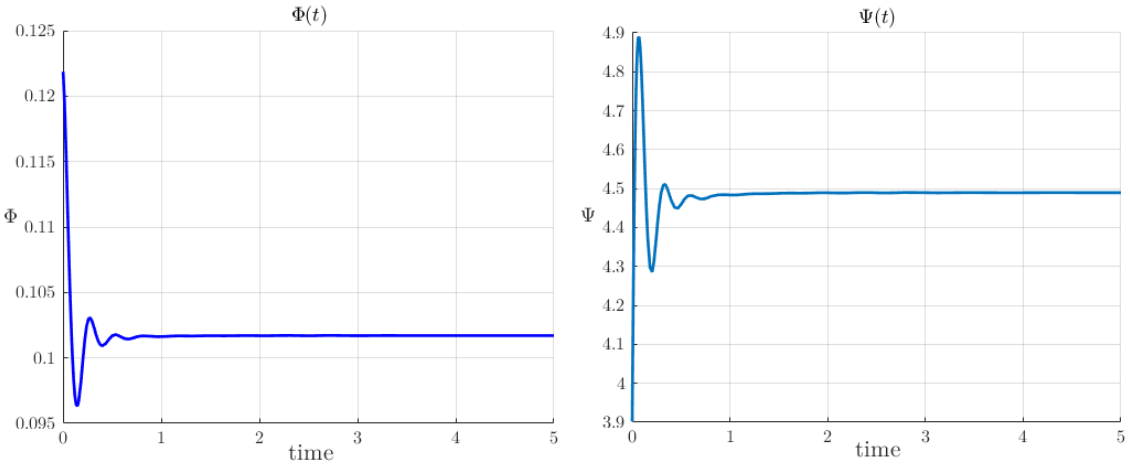


Figure 3.9: *Time evolution of nondimensional variables*

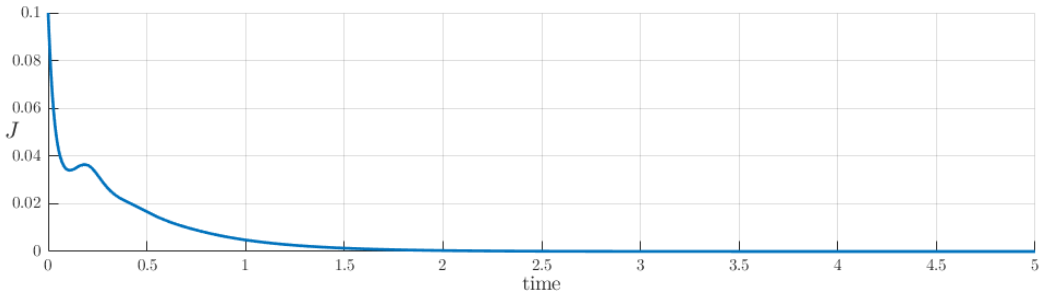


Figure 3.10: *Stall amplitude evolution*

3.3.2 Surge examples

By setting the operating point on the left side of the curve maximum, we move into surge region. We were able to recreate two examples od deep surge: the first is based on these entry parameters.

Parameter	*Value*
ϕ_0	0.1049
ψ_0	4.458
B	5
γ_T	0.0497
t_{end}	60000

The starting point corresponds to the operating point, and was chosen on the stable region of the map. One may see that the inputs correspond to those used in the first stall examples, with the difference that this time a much larger Greitzer's parameter was chosen. This had an interesting impact on the time values. Since large B significantly increase the time necessary for the surge limit cycle to emerge completely, we were required to choose a very high final simulation time. The results are shown in figure 3.11.

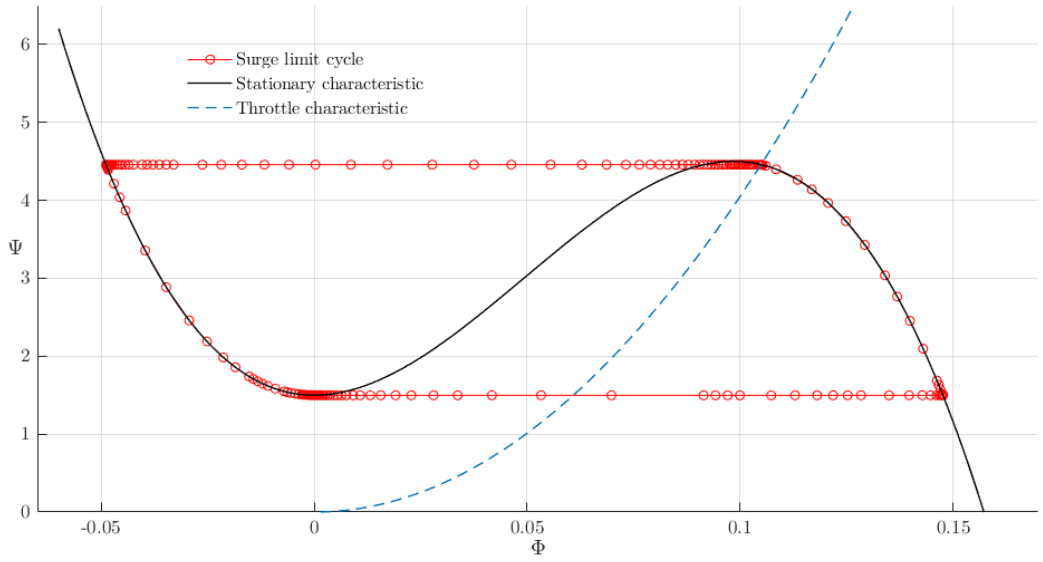


Figure 3.11: *Surge cycle from a stable operating point*

The limit cycle starts from the operating point, then develops as expected with jumps from one branch of the curve to the other as soon as stationary points are reached. As mentioned before, a whole cycle takes many seconds to form, and from the plot we may clearly see that the system has been actually brought into surge. Since the starting point was chosen beside the surge limit, we are now interested to see if stability is preserved. To do that, we might resort to the usual evolution of ϕ and ψ variables.

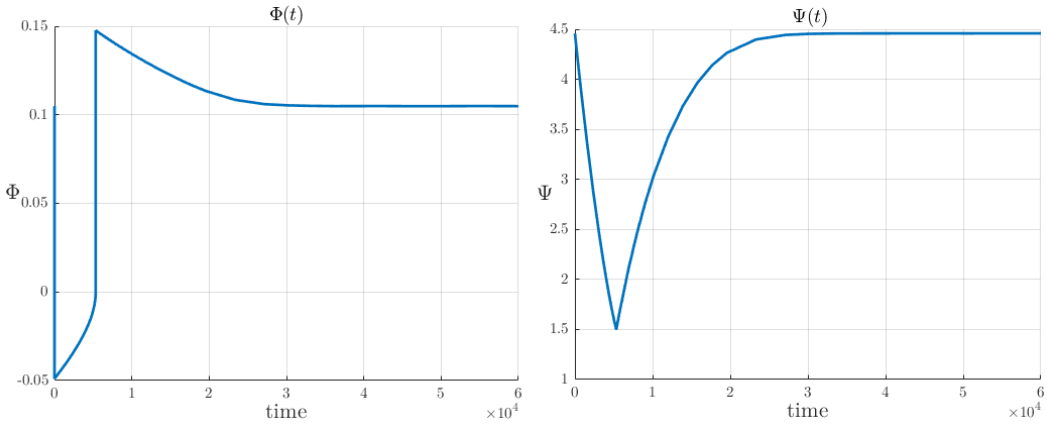


Figure 3.12: *Time evolution of nondimensional variables*

Figure 3.12 shows that after a single surge cycle the system stabilises on the operating condition, and then remains constant. This is the proof that, even starting from a stable point, we might actually fall into surge, but no limit cycle is produced. For surge simulations no $J - t$ plot is reported, since (as expected from the theory) the squared stall amplitude proved to be uniformly null.

In order to recreate a real surge cycle we need to move in the unstable section of the characteristic. That's what we did as a second case, for which the following entry values were set:

Parameter	*Value*
ϕ_0	0.07703
ψ_0	4.137
B	0.05
γ_T	0.0379
t_{end}	15

As in the previous case, the throttle gain was calculated from the starting point, which, as explained, lies on the growing branch of the curve. Since the system is now naturally prone to be unstable, a small B value was enough to produce surge, and consequently also the final time necessary was lower.

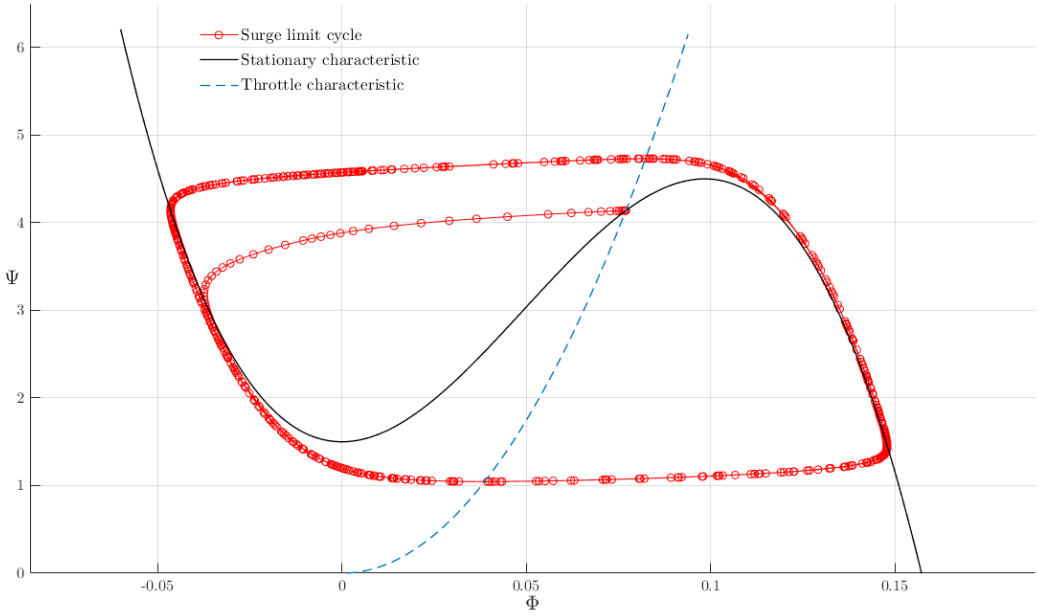


Figure 3.13: *Surge cycle from an unstable operating point*

The dynamic is evidently more extreme than in the previous case. The system starts from the operating point, and then rapidly evolves to create the broad cycle typical of surge. Compared to before, we may also notice that the cycle moves even beyond the extreme values of the characteristic, creating possibly more stressful conditions.

As in previous cases, the time evolution of the variables is a useful means of analysis. From figure 3.14 we appreciate the surge limit cycle. This time the system can no longer absorb the instability insurgence, and as a result this is indefinitely repeated over time in an endless loop.

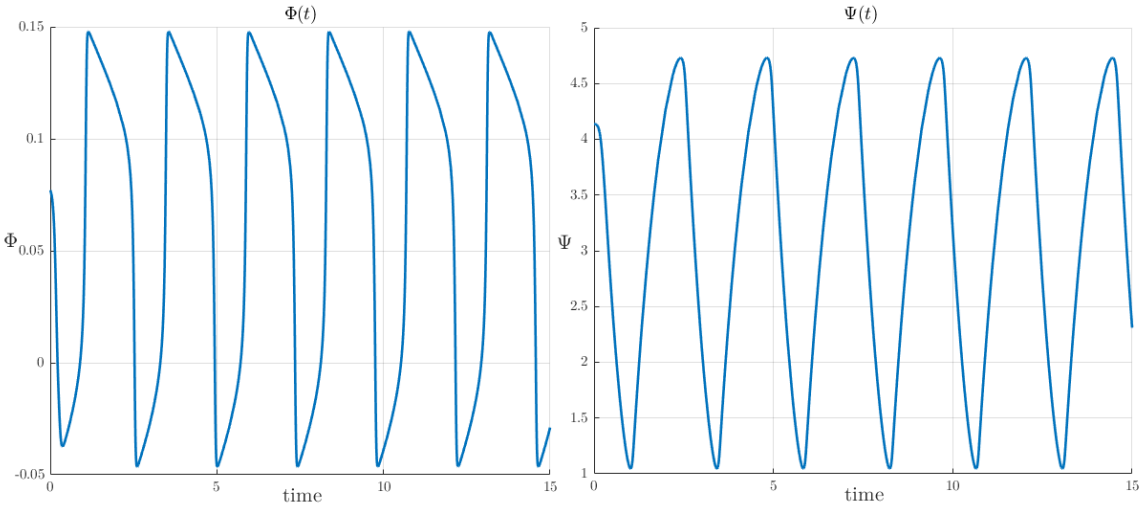


Figure 3.14: *Time evolution of nondimensional variables*

Chapter 4

Compressor map adaptation to the instability model

In this chapter we will try to manipulate that map to make it compatible with the function promoted by the Moore-Greitzer model (reported in 3.2), in a process that acts as a bridge with the following part of this work, which is introducing it in an engine model.

To do that, it is important to understand with clarity how that map is built, and what problems may come up when trying to reproduce it analytically.

4.1 Notation and main issues

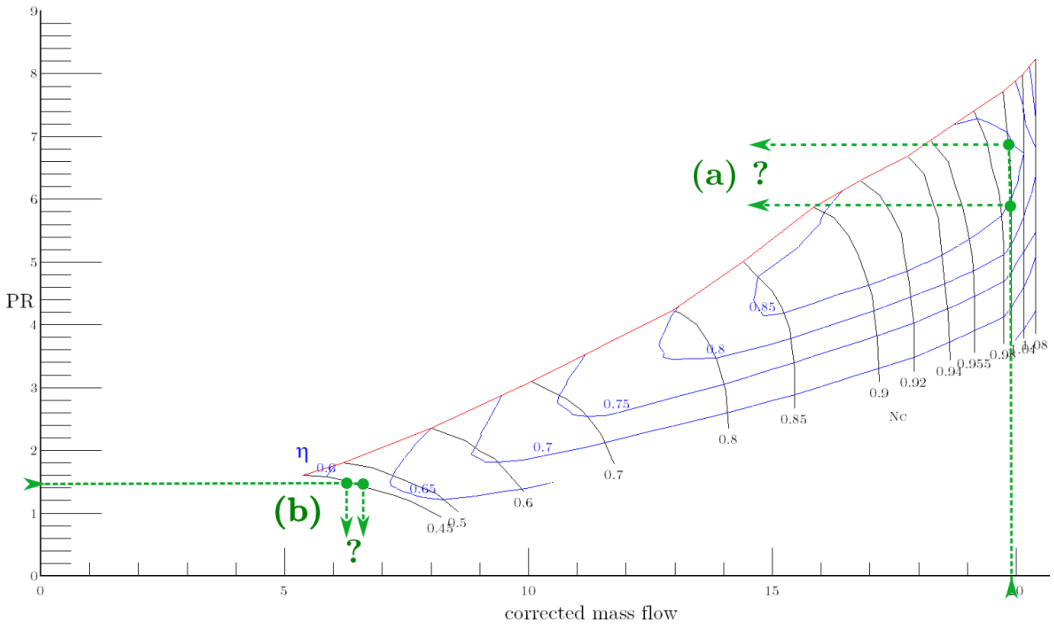


Figure 4.1: *Complete compressor map (GSP) and point detecting ambiguity*

The main reason for using a map to understand how the compressor works in every condition is the agility to read all the useful information at once from a single chart. As previously explained, the map relates three basic characteristics:

1. corrected mass flow,
2. pressure ratio,
3. corrected speed.

It is also possible to add a further parameter to the these last ones, which is **efficiency**. Usually it is represented in the form of level curves, and it is useful in the compressor design point selection.

Figure 4.1 shows all these parameters gathered together, and also points out one of the key issues of the compressor map. Observing the iso-speed lines one can immediately see that at high corrected speed the slope is almost **vertical**, whereas ah low speed it tends to become **horizontal**. This brings about significant difficulties in reading the points corresponding to entry values, either on the x-axis or the y-axis. For example, should it be required to assess the pressure ratio determined by a given corrected mass flow, it would be impossible to discern from two equivalent points **(a)**; if else from the pressure ratio we had to go back to the mass flowing in the machine **(b)**, a very small variation would produce much different results, and possibly big errors.

To solve this quite big inconvenience new reference lines have been added to the map, called **beta lines**. They are scaled from 0 to 1, and allow to translate the information reported on the map into matrices that are much easier to use, especially in those uncomfortable zones. In particular, by considering the coordinates given by the beta lines and the iso-speed lines, it is possible to create a new computational grid that avoids overlapping of points, and therefore allows to clearly identify the needed values. Below is an example of a compressor map with beta lines.

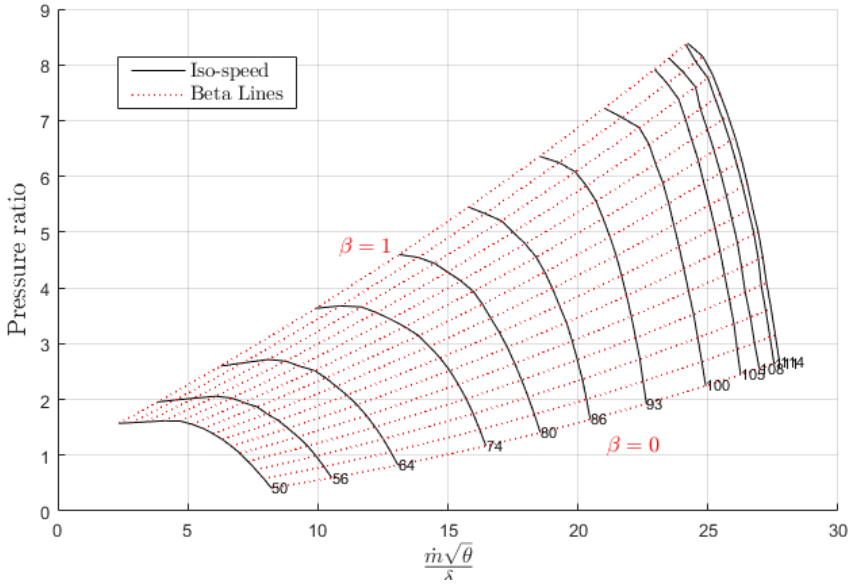
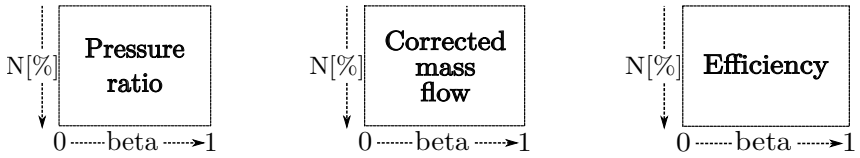


Figure 4.2: *Beta lines on the compressor map*

This kind of solution is employed by many commercial simulation programs like *GSP* [5], which reports the map values in the form of matrices like



It is important to point out that the concept of beta lines is very flexible, in the sense that their shape can vary according to the specific problem. This means that one can adopt any kind of expression for them, be it linear, parabolic or other, as long as the map discretization is effective. The model adopted in the present work is that of **parabolic beta lines**.

4.2 Interpolation procedure

Now that the issues related to the compressor map shape have been cleared out, it is possible to approach the main goal of this first section of the thesis: **interpolation**.

What we aim to achieve can be summed up in two points:

1. **approximating the experimental values** of the map (those on the right of the surge line) with an analytical function;

2. **extending** the characteristic **over the surge line** where no experimental point is available, creating a complete curve like the one reported back in figure 2.9.

Both these goals can be addressed by making use of the expression proposed in [2] and reported in the final section of chapter 2, namely the cubic polynomial that relates pressure rise coefficient and axial flow coefficient:

$$\psi_c = \psi_{c0} + H\left[1 + \frac{3}{2}\left(\frac{\phi}{W} - 1\right) - \frac{1}{2}\left(\frac{\phi}{W} - 1\right)^3\right]. \tag{4.1}$$

Clearly, before using this analytical function it is necessary to nondimensionalise the variables, which are given in their original form. At the end of this process, then, we will need both the complete dimensional compressor map and the nondimensional cubic characteristic.

To obtain this, two possible paths exist.

The first way to go by this problem is to make the approximation **before** the nondimensionalisation, thus obtaining a polynomial for every iso-speed line. The variables will be nondimensionalised later, so from a family of curves also the single characteristic will remain.

The second procedure instead consists in nondimensionalising the points straight away, and then a single interpolation will be sufficient to obtain Greitzer's cubic curve. At that point getting back to dimensional form will produce the complete approximated map.

Both of these approaches have been carried out in this thesis, so all the different results will be discussed and compared. Il will be seen that a simple interpolation, no matter how accurate the algorithm used may be, is not enough to obtain satisfactory results in terms of coherence with the experimental points. The reason for this resides again in the

unfavourable vertical trend of constant speed lines of the map, which is very hard to fit with the polynomial we use. [1]

Since approximation accuracy cannot be the only criterion in choosing one method rather than the others, different factors have been taken into account, like

- **readability** of the whole map;
- **respect** of the problem's physics (e.g. no negative values for the pressure ratio are acceptable);
- **practicality** of the interpolation algorithm for future use in simulations.

4.2.1 The engine model: General Electric J85-13 turbojet

Before outlining the actual interpolation procedures we briefly present the engine reference used for all the calculations. Our choice went for a model that is both common (and therefore well described in literature) and essential in its architecture, namely the **GE J85-13**.

The J85 is a small single-shaft turbojet introduced in 1950 mainly for military purposes and still used today, being considered one of General Electric's longest service engines. Its compressor is composed of eight stages and preceded by a variable camber IGV, as sketched below (figure from [7])

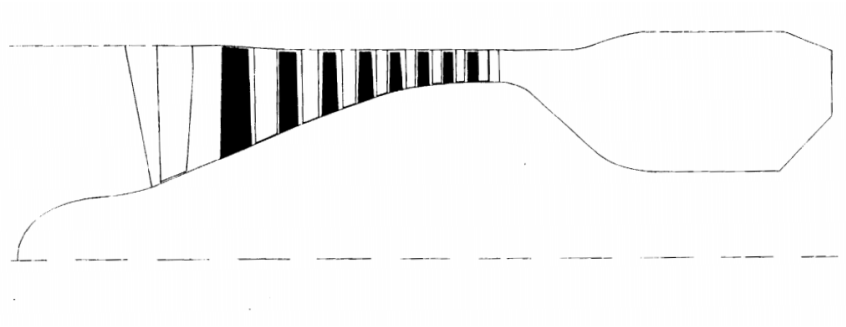


Figure 4.3: *Section of a J85-13 compressor*

All the data used for the original compressor map will be taken from the **GSP** software ([5]), whereas geometrical parameters and atmospheric conditions adopted in the calculations are reported in the table.

Parameter	*Symbol*	*Value*	*Unit*
Density at S.L.	ϱ_0	1.225	kg/m^3
Pressure at S.L.	p_0	100000	Pa
Mean radius	r_m	0.19685	m
Hub radius at entry	r_{0h}	0.09622	m
Tip radius at entry	r_{0t}	0.20193	m
Blockage coefficient	Q	0.97	
Design RPM	N_{des}	16540	

For a more comprehensive description of GE J85 see [8].

[1] For a more complex solution to this problem we refer therefore to [6].

4.2.2 The selected method: parametric interpolation

Among the many alternative solutions to our problem, which will be briefly discussed later on, we chose a method that seemed to be a good compromise between implementation simplicity and technical elegance.

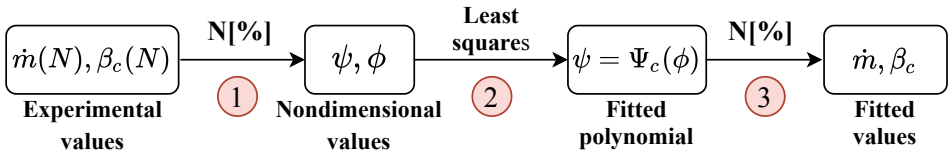
Parametric interpolation refers to a system that makes use of a single curve to fit all the experimental data in its nondimensional form, and then explodes it curve into a group of curves by means of a specific value, that of the **corrected speed**.

In other words, we used the implicit dependence of our nondimensional variables on the N parameter, as visible in

$$\psi(\phi, N) = \Psi_0 + H\left[1 + \frac{3}{2}\left(1 - \frac{\phi(N)}{W}\right) - \frac{1}{2}\left(1 - \frac{\phi(N)}{W}\right)^3\right].$$

and used it first to approximate the data, and then to get back to the dimensional map. The explicit process then consists of three steps:

1. the initial variables are nondimensionalised with their respective speed values;
2. the nondimensional data is fitted with a Greitzer characteristic (4.2) in the least squares sense;
3. the obtained third order polynomial is expansed into a group of iso-speed lines by reassigning the various corrected speeds.



To achieve this we used the basic definitions of nondimensional parameters

$$\phi = \frac{\dot{m}_c}{\varrho A_d U} \qquad \text{and} \qquad \psi = \frac{p_0}{\varrho U^2}(\beta_c - 1), \tag{4.2}$$

where the N dependency can be explicitly deduced from the tangential speed

$$U = \omega r_m = \frac{2\pi N}{60} r_m.$$

The core of this method, then, was finding the coefficients Ψ_0, H, W that gave the best approximation to the data, provided that a quite broad error was expected as a consequence of one polynomial fitting all the points at once.

Figure 4.4 reports the result of this process, showing the cubic characteristic that best approximates the usual set of data.

As mentioned, we cannot require this interpolation to be particularly accurate, and in fact the dashed line fits the values with a quite high relative residual. The real benefit, however, resides in the fact that the single interpolating function had the ability to evolve freely among the experimental points, and therefore be respectful of the problem physical constraints.

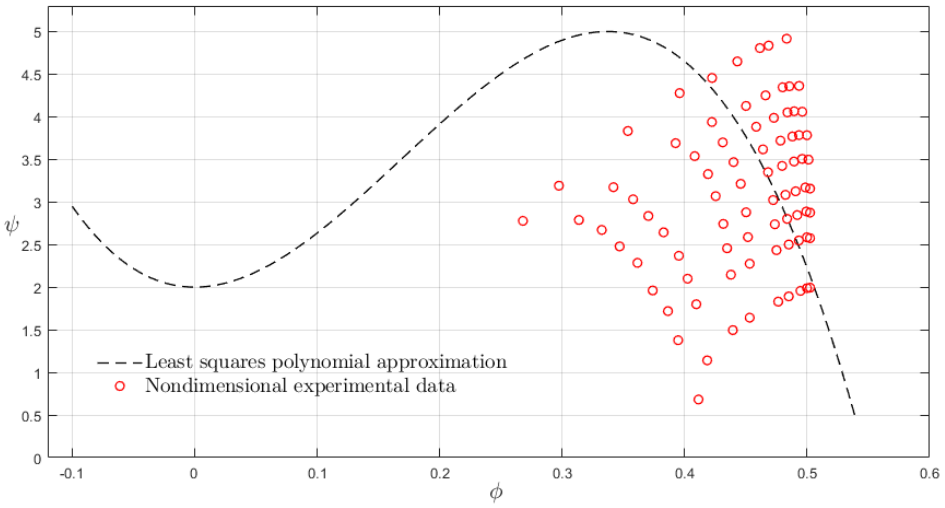


Figure 4.4: *Polynomial approximation of the nondimensional dataset.*

Furthermore, the advantages of this method immediately clear out as soon as we get back to the dimensional form of the variables. To do that it is necessary to invert the relations written in 4.2.2, so that from the $\psi - \phi$ plane we can move back to the complete compressor map. More specifically, first a range for the independent variable ϕ is established, then the corresponding ψ values are computed through the interpolating cubic function, and finally the corrected speed dependency is once again exploited to develop the polynomial into the usual set of iso-speed lines.

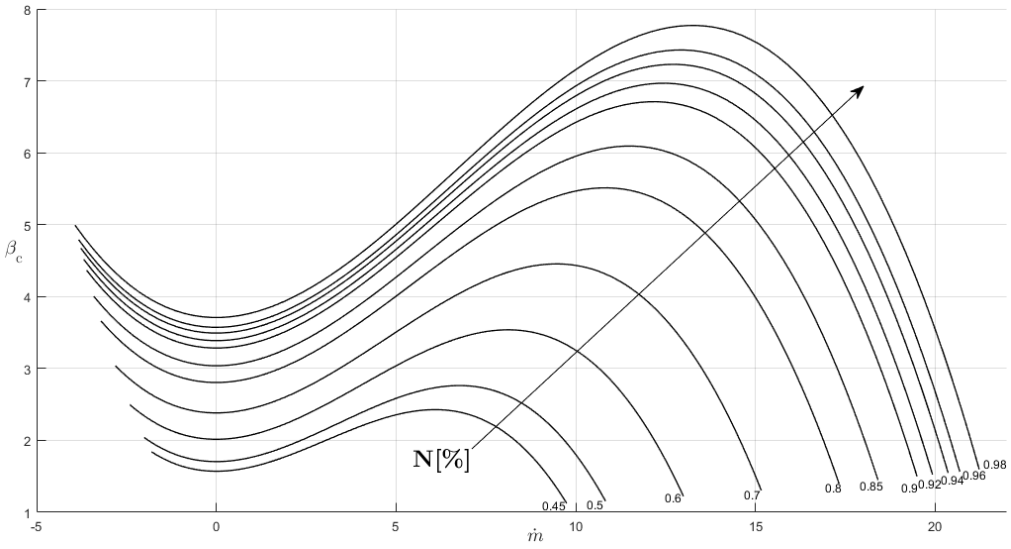


Figure 4.5: *Complete compressor map with parametric interpolation*

In figure 4.5 above the complete compressor map derived from parametric interpolation is reported. At first glance it is immediately evident that this solution really is a good

compromise of our prerequisites. Not only is the map coherent with the physical problem, being completely contained in the positive part of the β_c axis, but it is also smoothly distributed along the plane, producing no overlapping or crossing whatsoever. Of course the main drawback remains the poor accuracy of the approximation, as seen in figure 4.6, but yet as all the other requirements are met, we can easily accept that limitation. [2]

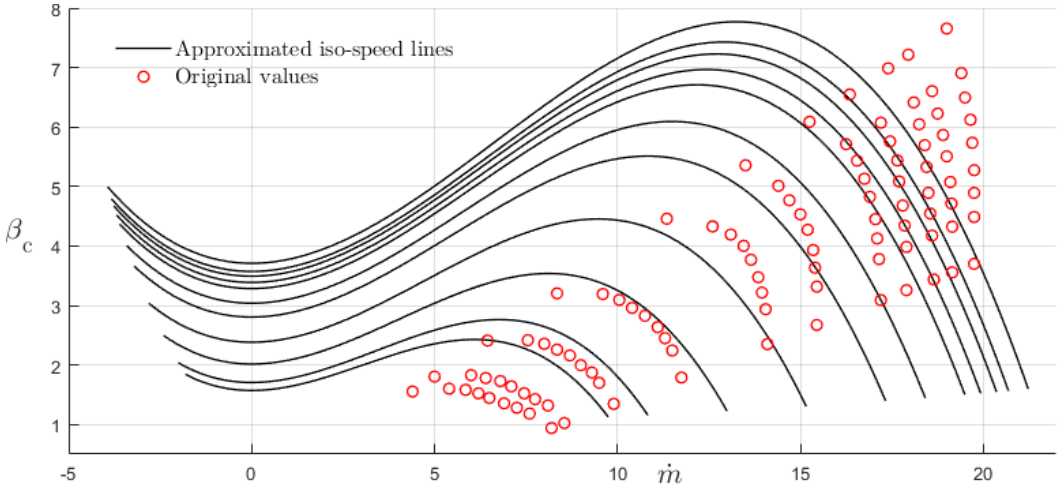


Figure 4.6: *Parametric interpolation accuracy limitation*

It is now worth mentioning that an attempt was actually made to tend to the poor accuracy of this solution. The idea was splitting the dataset into two groups, one for the **high speed** lines and one for the **low speed** ones, in order to obtain not one but two cubic characteristics in the nondimensional plane, so that both could follow more precisely their own set of points (see figure 4.7 below).

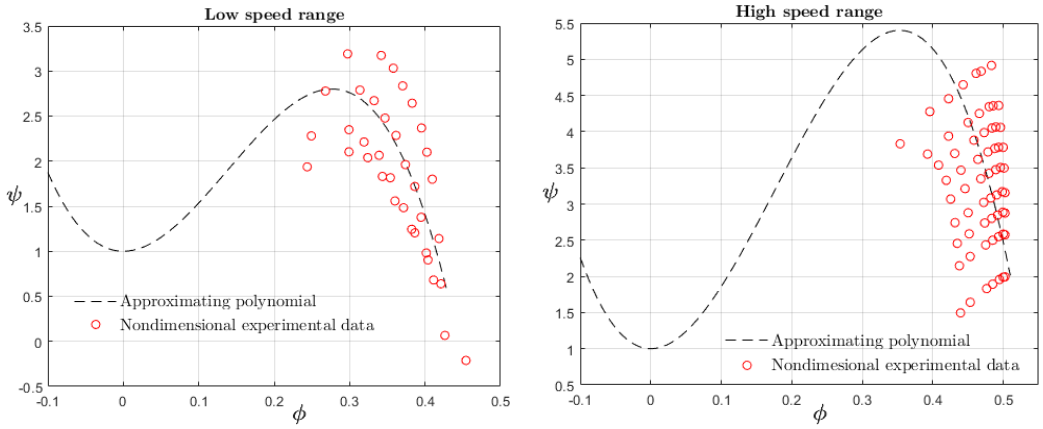


Figure 4.7: *Split speed range parametric interpolation*

[2] The aim of this thesis remains performing a functioning simulation of an engine through the Greitzer model, so the priority is given to the practicality and resilience of the interpolation rather than to its mathematical precision. Once the model does work, better approximation methods can easily be introduced.

Applying the same process discussed before, these cubic functions were then brought back in the dimensional plane and expanded in the various iso-speed lines. All the derived curves have been plotted in the same chart in order to obtain the whole compressor map.

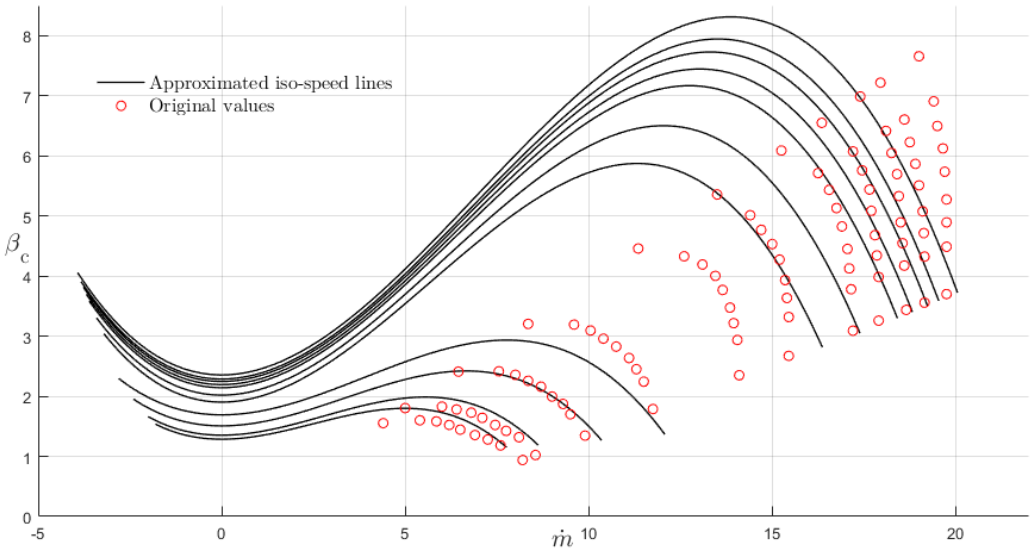


Figure 4.8: *Complete map with split parametric interpolation*

As seen above, this solution actually increases the method precision, especially at low speed ranges. Nevertheless, it was discarded because it didn't give a significant step up on the high ones, and at the same time produced a map that was not as clean and readable as the latter.

4.3 Alternative methods and their limits

4.3.1 Single iso-speed line fitting

The first attempt at compressor data interpolation was made in what was deemed the most direct approach. Every iso-speed line was singularly nondimensionalised according to its own tangential velocity U and than fitted with a third order polynomial in the form of

$$p(x) = ax^3 + bx^2 + c \tag{4.3}$$

using a polynomial regression model. The result was a family of approximated curves in the $\psi - \phi$ plane that seemed to accurately follow the experimental data trend, like shown here with a few examples.

The plots reported in figure 4.9 are related to just four nondimensional constant speed lines among the wide range that composes the original map [3]. Red dots represent the experimental points, whereas the black dashed lines correspond to the polynomial regressions in the form of 4.3.

It is immediately clear that the approximation is fine, as all cubic curves reproduce accurately the original data trends. However, problems arise as soon as we observe how the

[3] The compressor map of the J85 corresponds to the one presented in figure 4.1

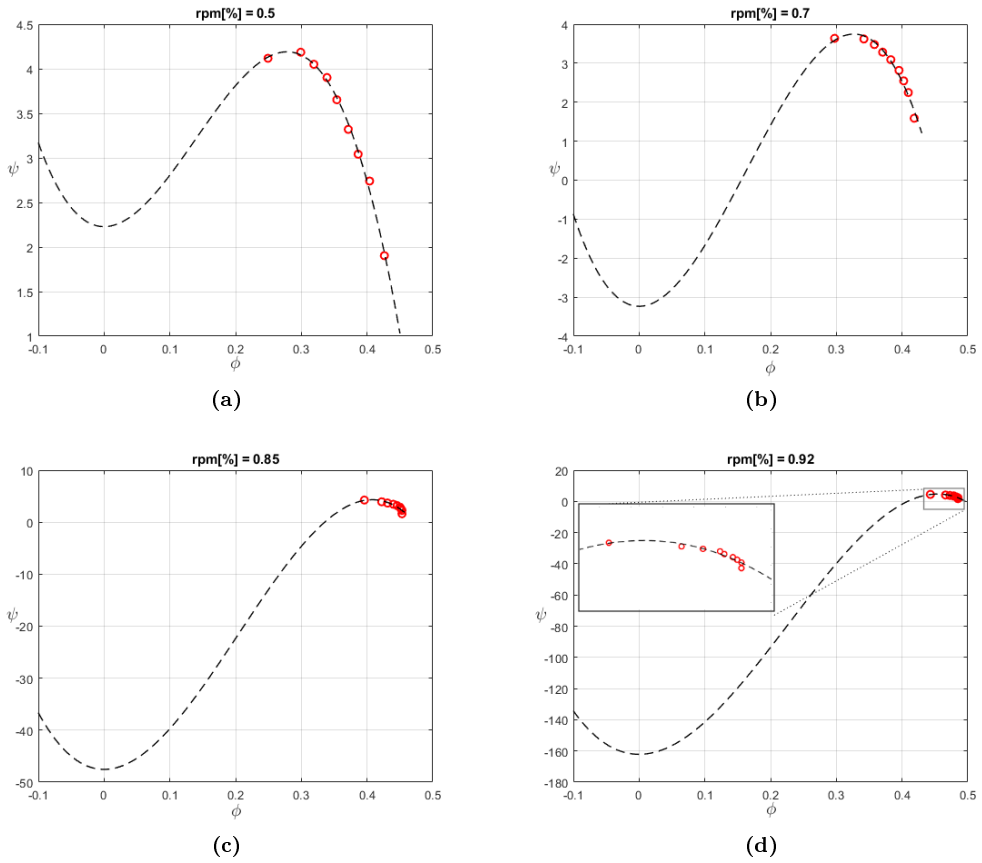


Figure 4.9: *Examples of results with first interpolation method*

interpolation behaves when the speed increases. The left part of the curves, which is related to the efficacy of post stall behaviour predictions, tends to diverge towards the negative values of the ψ axis, compressing the right part (the experimental one) into a negligible region of the map. This is not acceptable because the Moore-Greitzer model requires curves that are fully positive, as no pressure coefficient - as we defined it - can be lower than zero (it would mean a flow *expansion* is happening in the compressor). Of course this disappointing outcome is not dependant on a faulty interpolation method, but rather on the position of the data to be interpolated. In other words, if we demand the code to be both accurate in following the red dots and respect the constraints of the polynomial we chose (4.3), what we produce is a function that steeply becomes vertical and then negative.

Should we even come to terms with this deficiency, another issue - possibly bigger than the latter - emerges. When we transform the nondimensional characteristics back to their dimensional form and plot them all together, the result is not a coherent and readable map, but instead a chaotic overlapping of lines (figure 4.10).

A map like this is useless when introduced in a simulation because it determines ambiguity when reading values along either axis. Instead, what we need is a group of curves spreading independently on the plane without any crossing.

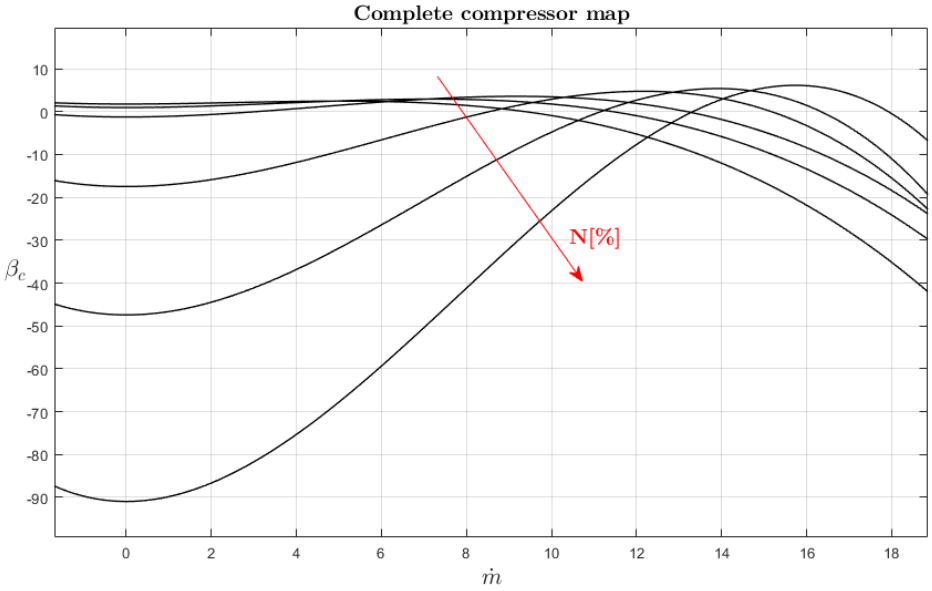


Figure 4.10: *Complete map and reading problems*

4.3.2 Bidimensional approximating surface

In order to systematically avoid any crossing a completely different approach can be applied. The idea was that the experimental points would now be fitted not with a curve but with a **surface**, in a tridimensional space where the third parameter would be given by the speed itself. This solution in very close to parametric interpolation, with the difference that now the parameter dependence was explicitly set from the outside, rather then looked for in the existent variables. From a simple cubic polynomial, the approximating function had now two variables, the mass flow coefficient and the speed, linked by some new constant parameters:

$$\psi = \psi(\phi, N, \text{constants}).$$

The core of this approach was then to find those parameters so that the associated surface best fitted the data in a **least squares** sense. Starting from the Greitzer polynomial reported in 4.2, we expressed the coefficients ψ_{c0}, H and W as a monotonic function of N. By doing so we made sure that, as the speed changed, every $\psi - \phi$ characteristic developed on a different plane of the space, therefore not interfering with the others. In other words, by adding a constraint to some key parameters we guaranteed that the system satisfied the needed topological conditions. This method was carried out with two different speed dependencies. The first solution corresponded to a so called **linear regression**, for which the coefficients had the form

$$\begin{aligned} \psi_{c0} &= a_0 + a_1 N \\ H &= b_0 + b_1 N \\ W &= c_0 + c_1 N, \end{aligned}$$

and thus the surface was described by the equation

$$\psi(\phi, N) = (a_0 + a_1 N) + (b_0 + b_1 N)\left[1 + \frac{3}{2}\left(1 - \frac{\phi}{c_0 + c_1 N}\right) - \frac{1}{2}\left(1 - \frac{\phi}{c_0 + c_1 N}\right)^3\right].$$

What we expect here is that every section along the $\psi - N$ plane contains the nondimensional characteristic associated to that specific speed, and that as a whole the characteristics linearly develop along the N axis. The result is shown in figure 4.11.

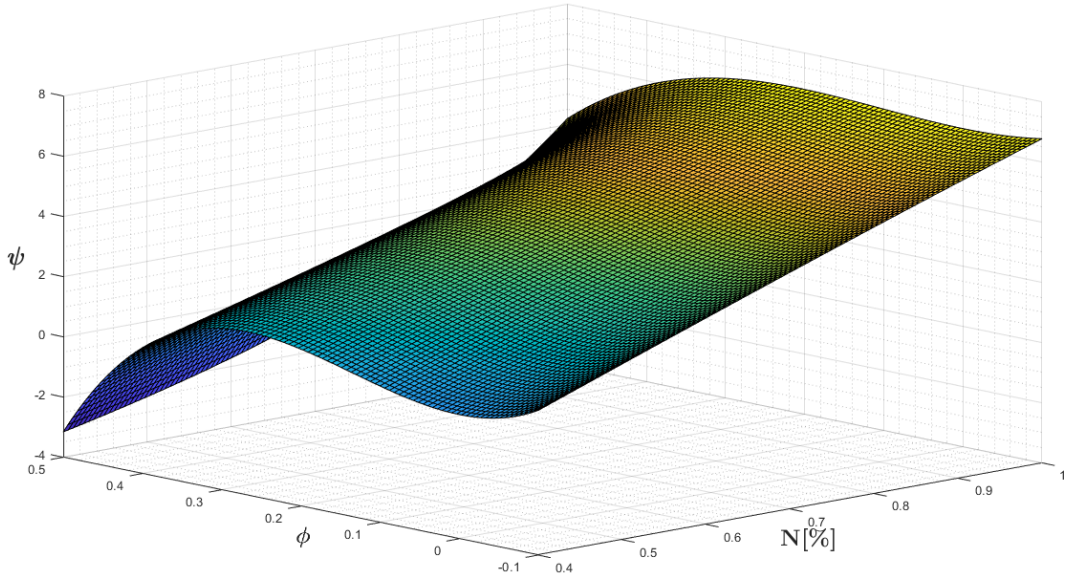


Figure 4.11: *Approximating surface with a linear regression*

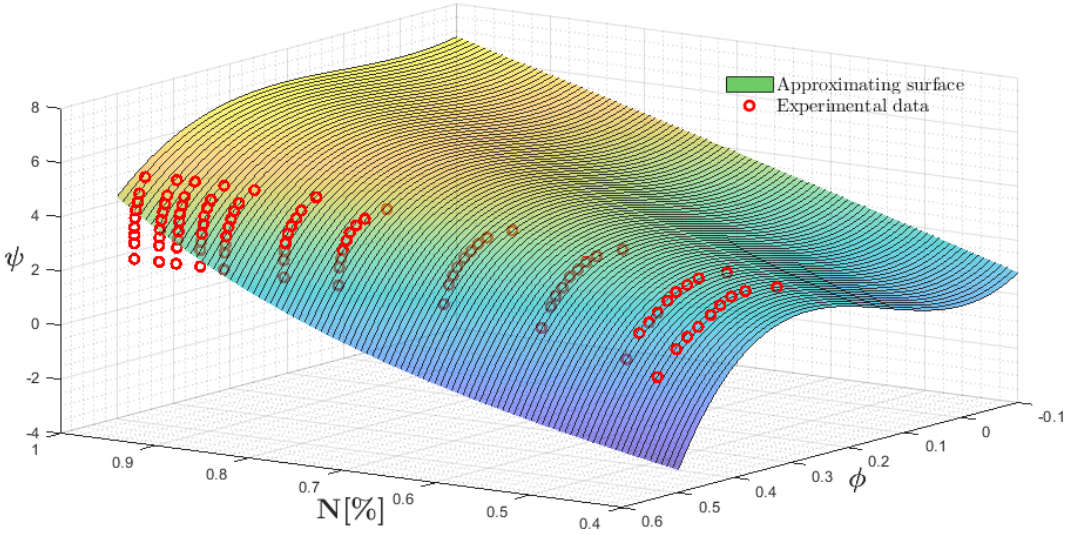


Figure 4.12: *Approximating surface with experimental point scatter*

For a better view of the approximation we can also visualise the experimental points over the surface, as shown above.

Figure 4.13 reports the same plots depicted in 4.9, only this time they are vertical sections of the surface above. Overall the approximation is less accurate than the one we previously

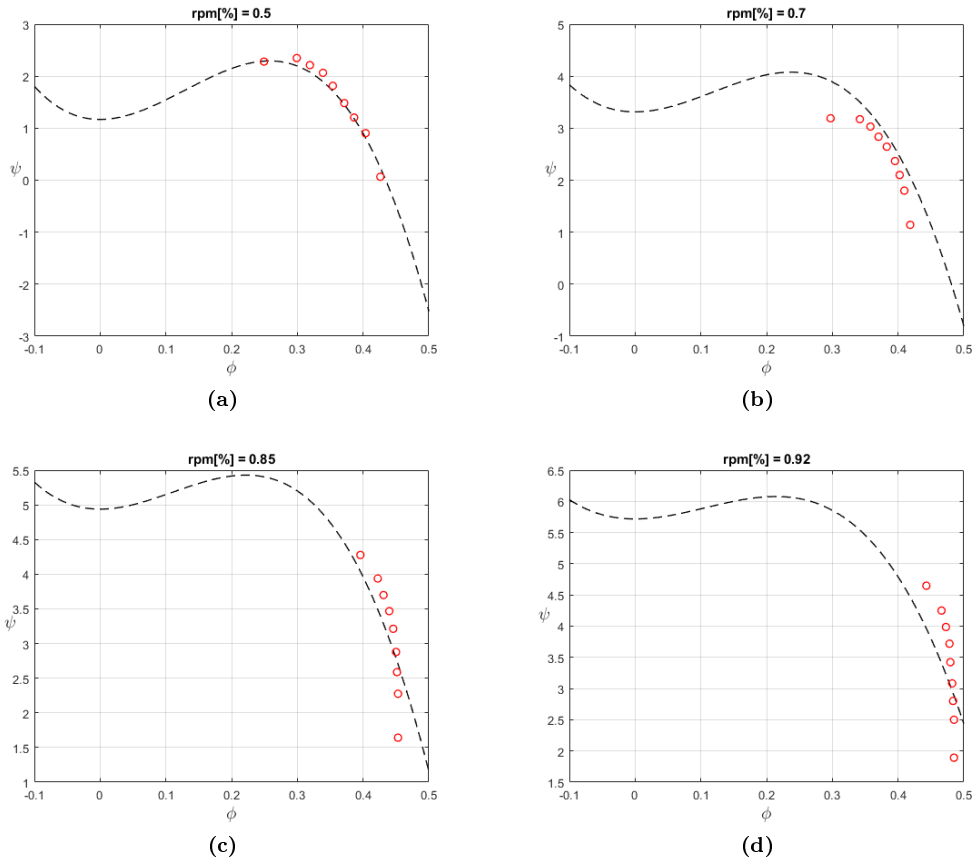


Figure 4.13: *Examples of results with second interpolation method*

obtained, and when the speed grows the interpolation gets progressively worse just like before. However, the shape of these curves is closer to the physical phenomenon: each one of them has a positive minimum, and no negative value is measured. Nevertheless, the main reason to prefer this method to the previous one, in spite of a greater error [4], goes back to its original purpose: avoid curves overlapping.

If we return to the dimensional form of each parameter, the compressor map obtained with this second approach to our problem is the following:

Evidently the one reported in figure 4.14 looks much more like the kind of compressor map we need. No crossing between iso-speed lines can be seen, and as a whole the map is clear and readable.

As previously mentioned, this method was conducted also in a second variant, that is using a **quadratic regression**. This time the interpolation parameters were defined as

$$\psi_{c0} = \alpha_0 + \alpha_1 N^2$$
$$H = \beta_0 + \beta_1 N^2$$
$$W = \gamma_0 + \gamma_1 N^2,$$

[4] The calculations showed that the squared sum of the 2-norm residuals for this interpolation was around 28, which is large but yet acceptable.

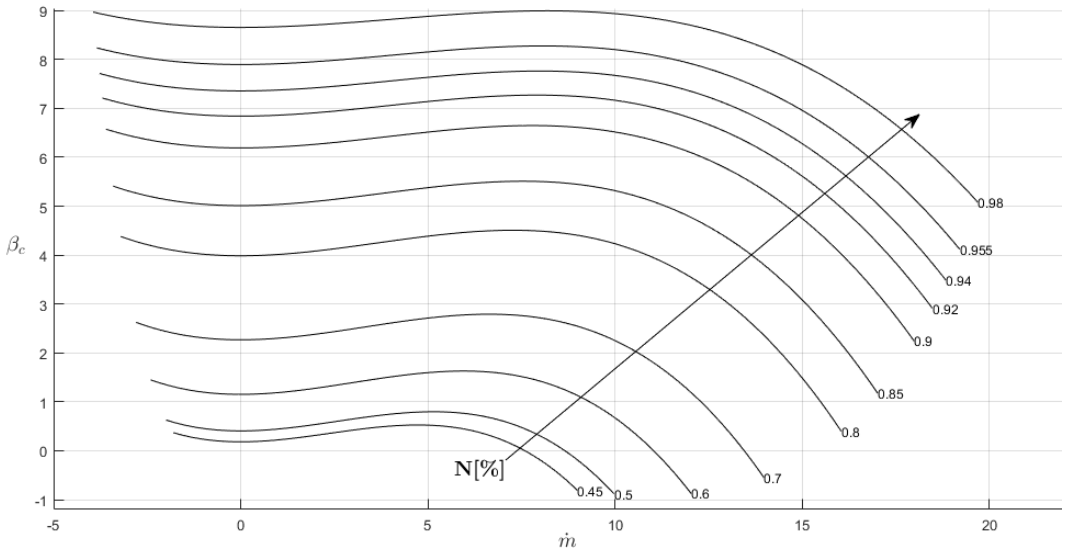


Figure 4.14: *Complete map with second interpolation method*

and the surface equation becomes

$$\psi(\phi, N) = (\alpha_0 + \alpha_1 N^2) + (\beta_0 + \beta_1 N^2)\left[1 + \frac{3}{2}\left(1 - \frac{\phi}{\gamma_0 + \gamma_1 N^2}\right) - \frac{1}{2}\left(1 - \frac{\phi}{\gamma_0 + \gamma_1 N^2}\right)^3\right].$$

The new surface appears much more complicated, due to the fact that the speed values are now distributed along a parabola and not a line.

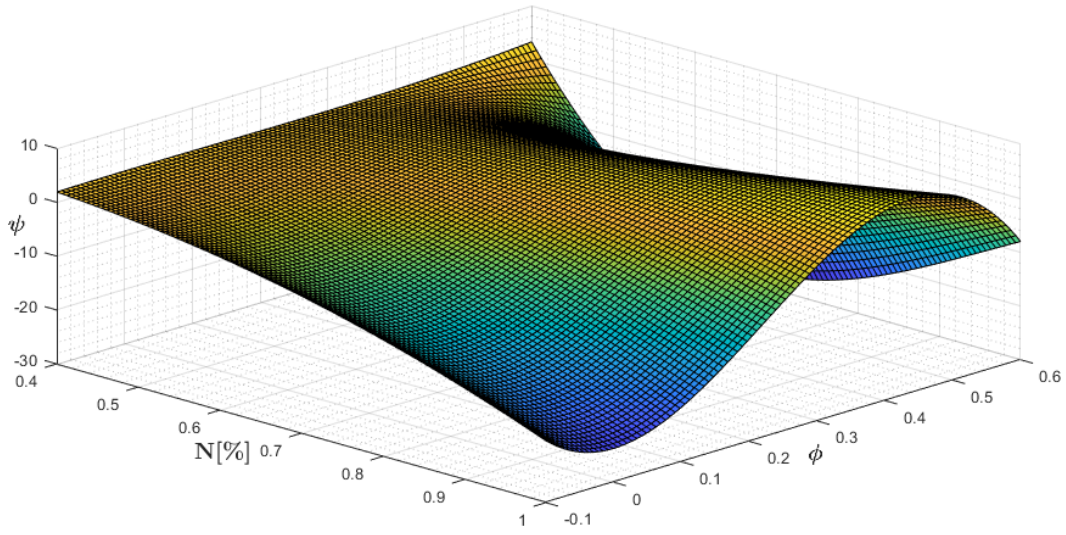


Figure 4.15: *Approximating surface with a quadratic regression*

Adding the data scatter to the plot as before we get an idea of the new solution accuracy.

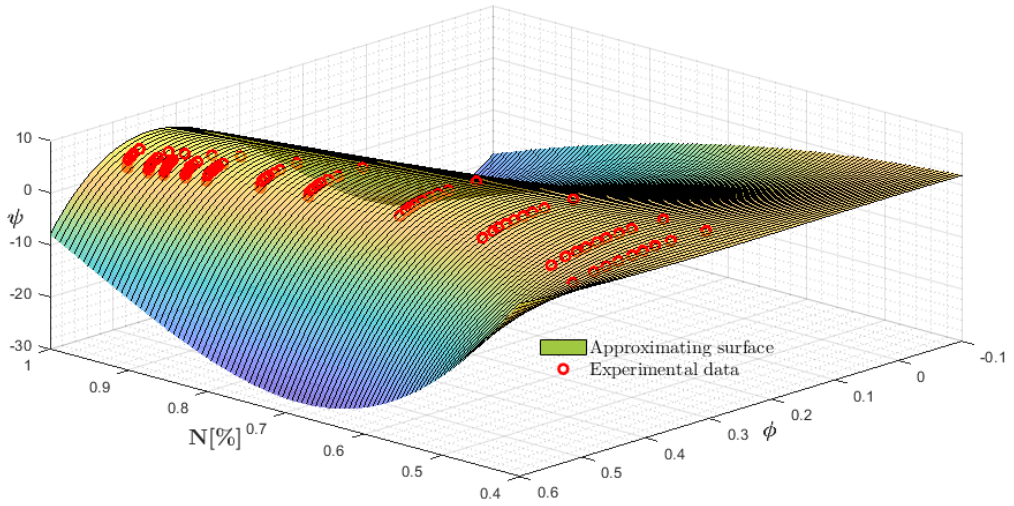


Figure 4.16: *New approximating surface with experimental point scatter*

This new type of regression improves the method accuracy , but to visualize this better let's resort to the usual set of examples:

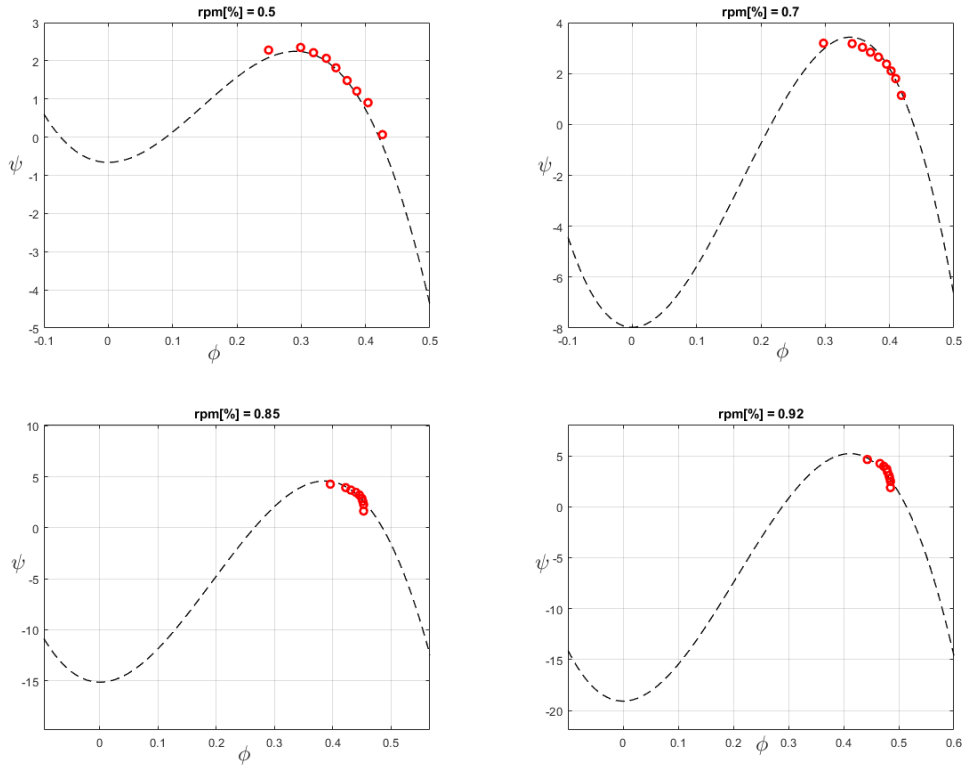


Figure 4.17: *New examples with quadratic regression interpolating surface*

The red dots are visibly better fitted by the dashed lines in this second variant than in the former one. Yet, with a keen eye it is possible to recognise a familiar pattern that relates figure 4.17 with the corresponding figure 4.9. In particular, we may see that the better approximation derives again from the curves tendency towards steep slopes, approaching negative values as soon as the speed increases. A direct consequence for this is that the region where the experimental data lie is only a minor section of the complete curve, giving predominance to the remaining part which instead is purely hypothetical.

A further confirmation that this last interpolation does not meet the requirements is given by the map plot (figure 4.18). The iso-speed lines are overlapped once again, showing that using a quadratic regression constraint is not sufficiently strict to prevent this specific issue. Interestingly enough, the analogies between this kind of interpolation and the one we tried in paragraph 4.3.1 are clear: this type of result is evidently what would fit best the original data, but unfortunately it is not physically acceptable.

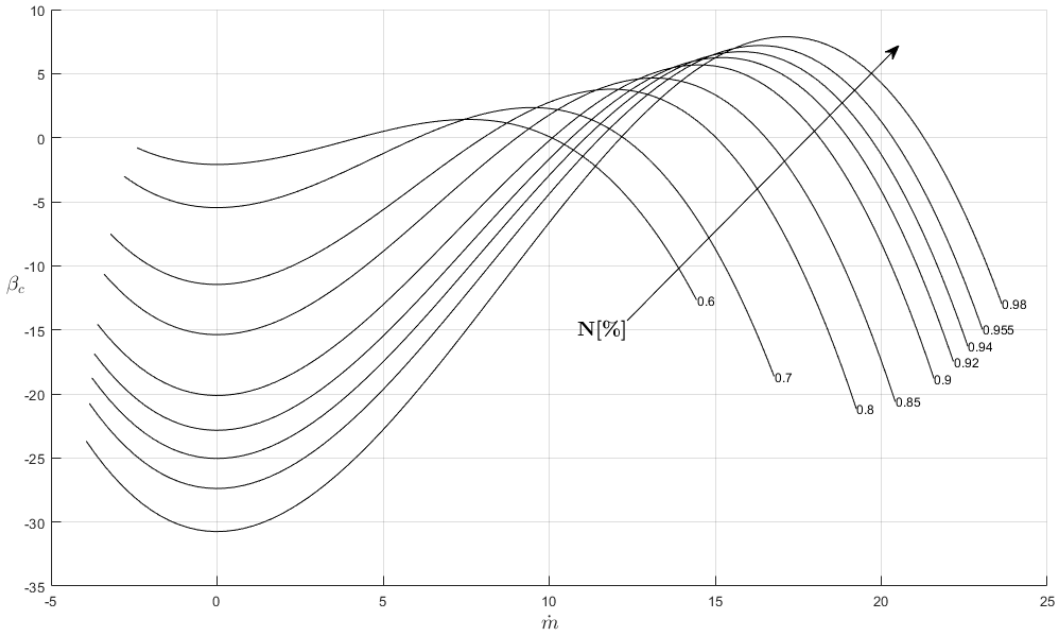


Figure 4.18: *Complete map with quadratic regression and similarities with first method*

4.4 Considerations

Different interpolation methods have been carried out to set the basis for this thesis, each with its own pros and cons.

Among all those, however, we deemed that the first we introduced was the most suitable for our purposes, both in terms of computational practicality and of the clear and readable graphical results. With the adoption of parametric interpolation the whole compressor model can be summarised in a single equation, which allows us to be independent from the discreet experimental data and gives us the huge possibility to determine analytically any point we need on the chart. The benefits of this ability is priceless because it quickens the calculations and guarantees robustness in the solutions.

As previously mentioned, Drummond *et al.* (2009) perfected this method using a more rigorous mathematical approach, obtaining maps that fitted the experimental data with far more accuracy. An example is reported below, where efficiency curves are also represented.

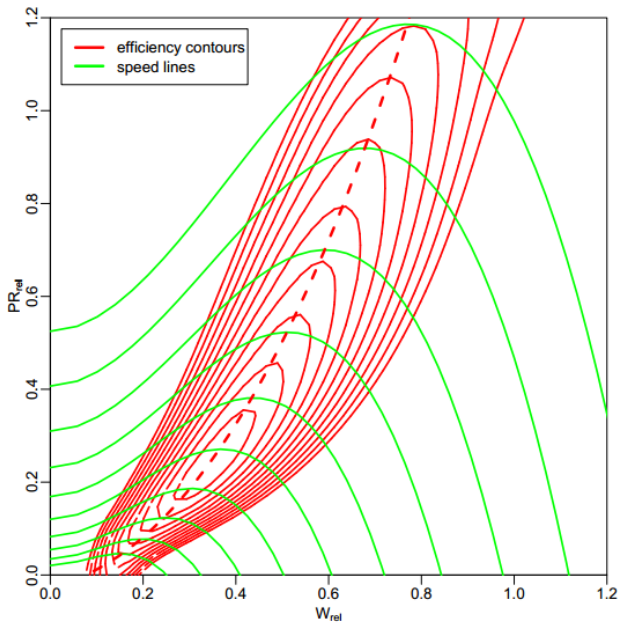


Figure 4.19: *Picture taken from [6], page 4*

Chapter 5

Engine simulation with extended compressor map

In this chapter we will get closer to the main core of this thesis, which is performing a simulation of the whole J85 engine. The basis for this process has been achieved by using the previous works by [9] and [10], who developed the simulation code for a general engine and for a sub-idle working engine respectively. The main upgrade we made to those works, other then integrating Greizter's model - as broadly explored in the next chapter -, has been introducing the approximated compressor map obtained early on.

In the following pages we will outline how such a simulation works, with the help of the flowcharts that guided the programming process. More specifically, three phases will be discussed:

1. **On Design** performance
2. **Off Design** performance
3. **Transient** response.

5.1 Design Point and performance

The **design point** is the thermodynamic and mechanical condition where the engine works nominally, that is when it satisfies the original requirements. Usually this configuration represents the optimum on the Joule-Brayton cycle, and is chosen because it guarantees the best compromise possible among the values (still in the pre-surge region) of

- efficiency,
- pressure ratio,
- mass flow rate.

The engine is supposed to work on design for the most part of its operation, and the design point is usually visualized on the compressor map. The equations that define all the design parameters are left to the reader, and can be easily found in [10] or [11]; here we will simply report the most significant ones.

Parameter	*Symbol*	*Value*	*Unit*
Rotating speed	N_{des}	16540	rpm
Corrected rotating speed	$N[\%]$	100	
Compressor mass flow rate	$\dot{m}_{c,des}$	19.90	kg/s
Fuel mass flow rate	$\dot{m}_{b,des}$	0.38	kg/s
Compressor pressure ratio	$\beta_{c,des}$	6.92	
Turbine pressure ratio	$\beta_{T,des}$	2.442	
Compressor efficiency	$\eta_{c,des}$	0.8686	
Turbine efficiency	$\eta_{T,des}$	0.9292	
Turbine inlet temperature	T_4°	1332	K
Compressor power	P_c	4895.9	kW
Turbine power	P_T	5154.6	kW

The visual representation of the design point on the approximated compressor map is then shown in figure 5.1.

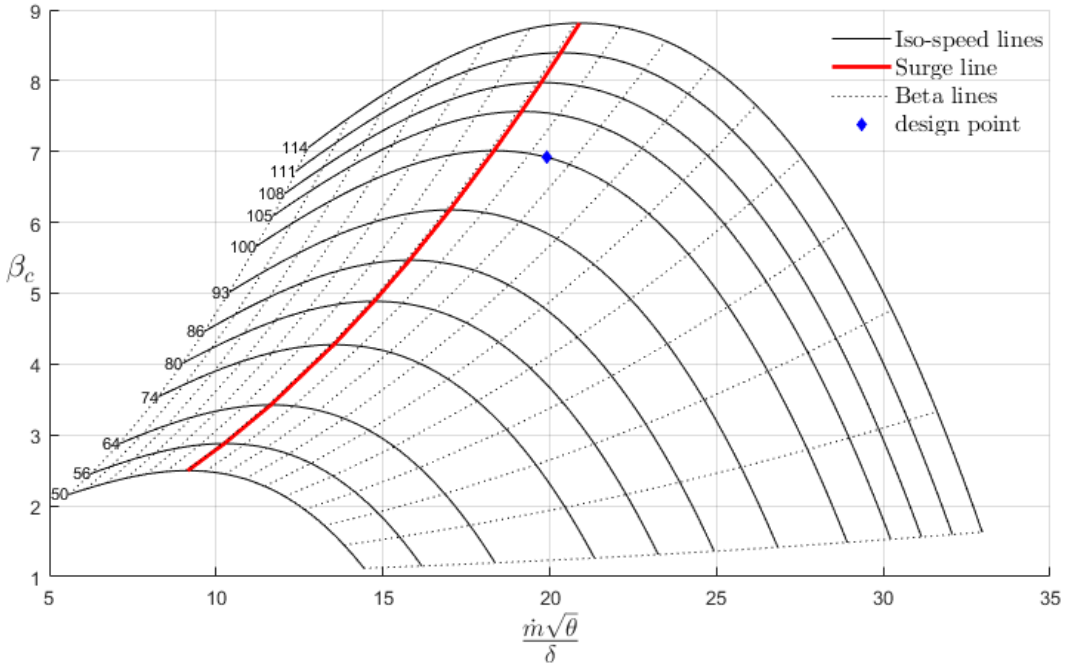


Figure 5.1: *Design point on compressor map*

5.2 Off Design and steady-state line

Off Design performance describes how the engine works the moment it departs from nominal conditions. Of course, not any random point on the compressor map can potentially be a working configuration, as some essential requirements have to be satisfied in order for the machine to operate properly. In the present work, we chose to evaluate off design conditions by varying the fuel flow from a maximum corresponding to the design value, down to a

minimum set to $\dot{m}_{b,min} = 0.054\,\mathrm{kg/s}$. Both turbine and nozzle were assumed to be choked. Three determinant requisites were then demanded to determine every **steady state point**:

i. *mass flow equivalence* in compressor and turbine, where $\dot{m}_T = \dot{m}_c + \dot{m}_b - \dot{m}_{bleed}$

ii. *mass flow equivalence* in nozzle, where $\dot{m}_N = \dot{m}_T + \dot{m}_{bleed}$

iii. *power balance*, where $P_T = \frac{P_c}{\eta_c}$.

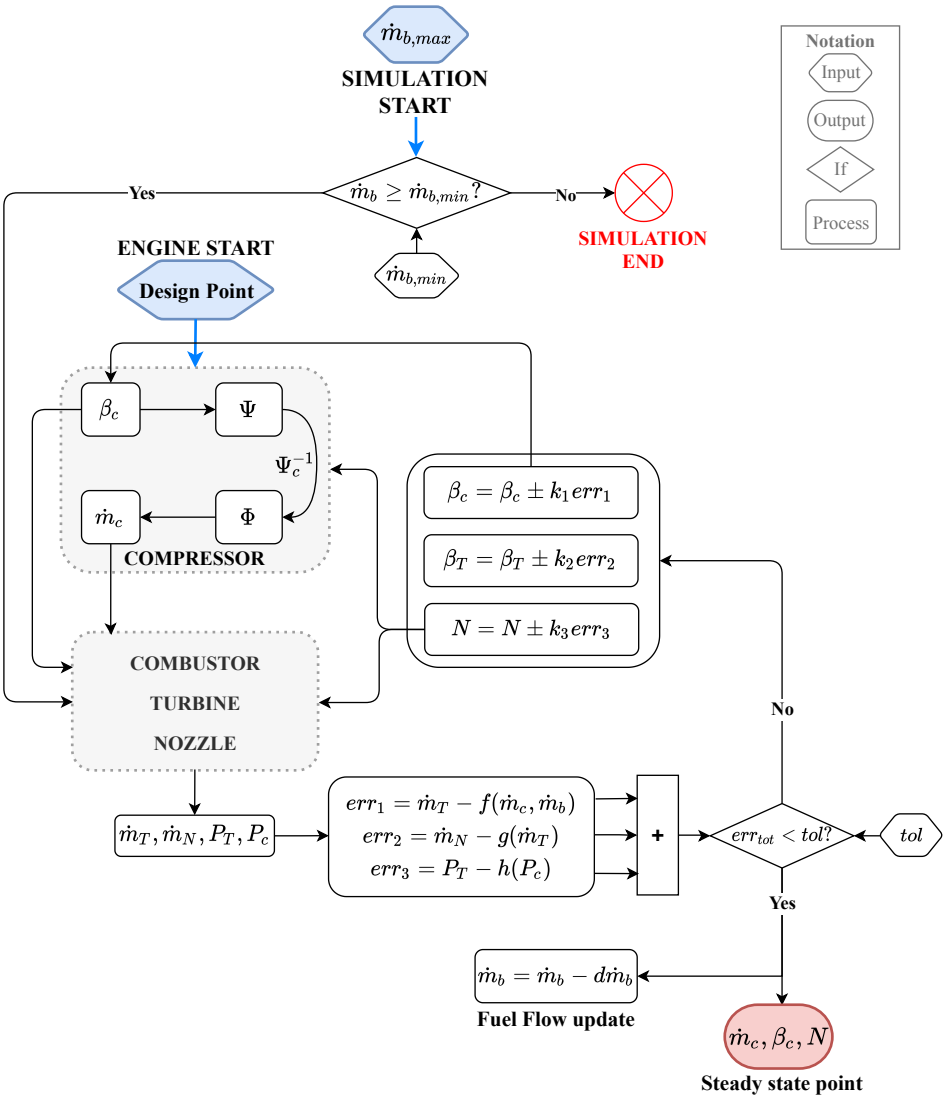


Figure 5.2: *Off Design simulation flowchart*

Figure 5.2 represents the flowchart used in the steady-state simulation development. The code is based on two nested cycles: the outer one handles the effect of the throttle through the fuel injection variation; the inner one computes, for every fuel flow value, the engine parameters (like $\dot{m}_c, \beta_c$ and N) that satisfy the three conditions above.

The simulation receives as initial input the design values listed in the table above, together with an established amount of fuel flow. For that specific $\dot{m}_b$ an **error-controlled regulation** feedback is used to converge towards the steady-state point, which is the final output of every loop. The fuel flow is then reduced (we assume that the engine is being slowed down), and the operation is repeated to obtain a new point.

If we report all the steady-state points obtained from the simulation on the compressor map, we obtain the **steady-state working line**, as shown in figure 5.3.

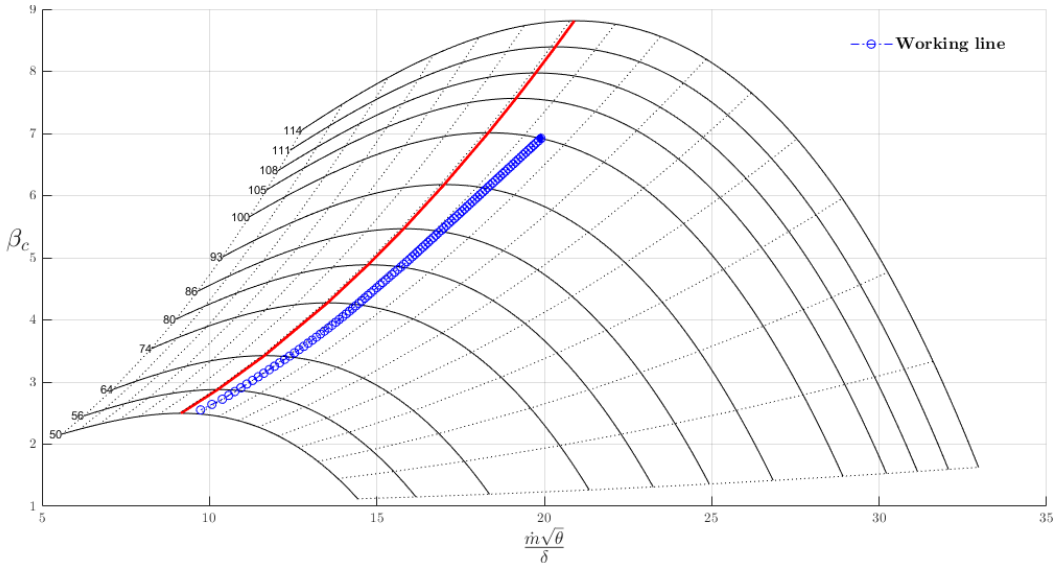


Figure 5.3: *Steady-state working line*

To guarantee that the engine operates in a safe region of the map, it is necessary that the working line keeps a certain distance from the surge line. A quantification of such distance is given by the **surge margin**, that represents the relative distance of a point on the working line (WL) from the correspondent on the surge line (SL). Among the many definitions of this quantity, the one we adopted is the following:

$$SM = \left(\frac{\beta_{c,SL}}{\beta_{c,WL}} - 1 \right) \times 100. \tag{5.1}$$

Applying 5.2 to our specific case study, we get a decent minimum surge margin value of 11.73% at about the middle of the working line.

5.2.1 Compressor model integration

Now that the general outline of the Off Design simulation has been given, it is important to get into more detail about how the new compressor model described in chapter 4 has been integrated in it.

As mentioned before, whereas the original compressor map was based on discreet data derived from experimental tests, the new map has been shrunk into a single analytical function, the cubic characteristic. This means that the simulation does not need to perform

interpolations anytime an intermediate value in required, but it just needs to evaluate that function.

Based on the block of flowchart 5.2 devoted to the **COMPRESSOR**, at every loop the component model receives a value of β_c as an input, be it either the design compressor ratio at the simulation start or the updated one coming from the feedback branch. The parameter is then nondimensionalised to move from $\beta_c - \dot{m}_c$ to $\psi - \phi$ plane: this operation, which required also the notion of the corrected speed N, was conducted through the usual relation:

$$\psi = \frac{p_0}{\varrho U^2}(\beta_c - 1).$$

From the pressure coefficient we moved back to the flow coefficient through the compressor characteristic $\Psi_c(\phi)$, by computing the preimage(s) associated to the ψ entry. The problem with inverting a cubic function is that it is not bijective in the entire domain, so to certain values of ψ may correspond up to three real roots on the ϕ -axis (figure 5.4).

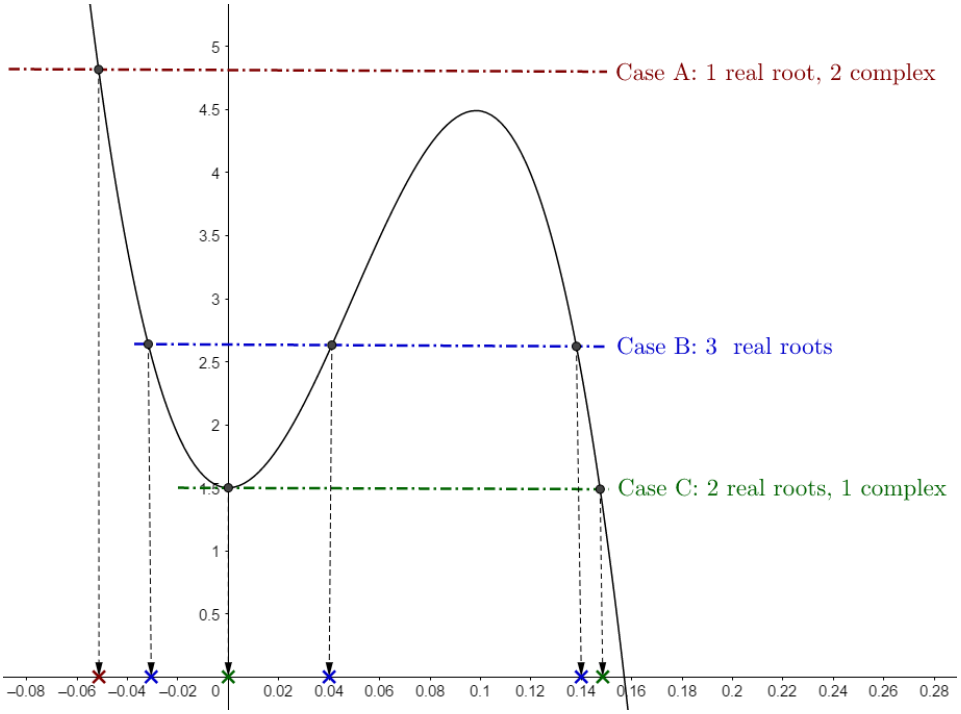


Figure 5.4: *Roots finding on compressor characteristic*

To tackle this issue we decided that any time multiple roots were identified, the one closer to the ϕ found in the previous iteration would be picked. This way the code was coherent and unexpected jumps on the map characteristic were avoided.

Considering that the purpose of our simulation is analysing the engine also in the unstable realm, however, a particular attention was necessary during the roots finding process. Usually numerical roots finding algorithms are predisposed to always compute all the solutions (three in our case), being them **real or complex**. Should the compressor work in a surged condition, the ψ in input might not be included in the region between the two stationary points of the cubic curve, so we may have only one real solution accompanied

by two complex conjugate ones. In this specific case, then, when choosing the correct ϕ we must select **the real one**, independently on the distance from its former value.

Once the correct flow coefficient was determined, we just needed to dimensionalise it to obtain the final output of the model, which was the mass flow rate:

$$\dot{m}_c = \varrho A_d U \phi.$$

5.3 Transient performance

Transient behaviour is of paramount importance in an engine simulation because it takes into account the effects of time evolution upon the machine dynamics. If Off Design analysis only determines steady state operating points, any time non-stationary phenomena are involved it is necessary to assert their influence on the problem physics, and therefore on the model equations.

A complete characterisation of these phenomena is quite difficult and would excessively complicate our simulation, so we refer to [12] for a thorough background on the topic. For our part, we will operate some major (but yet not so far-fetched) simplifications based on the relative magnitude of the times involved, putting ourselves in the position to neglect some of them. First though, we need to point out that the most important transients occurring during engine operation are:

- spools and rotors *inertias*;
- fluid dynamics *signal transmission*;
- *heat exchange* and *chemical reactions* in burner.

Spool dynamics require time-spans of about seconds to develop completely, whereas both fluid dynamics and chemical transients are assumed to happen almost instantaneously, surely in the sub-second realm. This means that the effects of the latter ones are long extinguished before the first one affects the simulation. We chose for this reason to consider them negligible, and to focus our attention on rotating parts inertia alone.

All kinds of transient maneuvers determine a power imbalance in the engine, undermining the subtle equilibrium that exists between the amount produced by the turbine and what is required by the compressor to move itself and the accessory drive. The moment we impose a quick variation in the spool rotating speed, be it an acceleration or a deceleration, the whole system will have to deal with either a lack or an excess of power due to the effect of inertias. To be more specific, considering the definitions

$$\begin{aligned} C &= \dot{\omega} I_P \\ P &= C\omega, \end{aligned} \tag{5.2}$$

we can assert that in case of power excess, the rotating speed instantaneous variation is

$$\dot{\omega} = \frac{\Delta C}{I_P} = \frac{C_T - C_c}{I_P} = \frac{1}{\omega I_P}\left(P_T - \frac{P_c}{\eta_m}\right) \tag{5.3}$$

Rotation speed is then the independent variable in transient simulations, and at every temporal step equation 5.3 will be integrated with a **successive substitutions method** in order to update the current N value. This solution was implemented in a similar code to

5.2, with the difference that both the fuel schedule and the steady-state working line were now given as an input.

The new flowchart that describes the model is shown in figure 5.5.

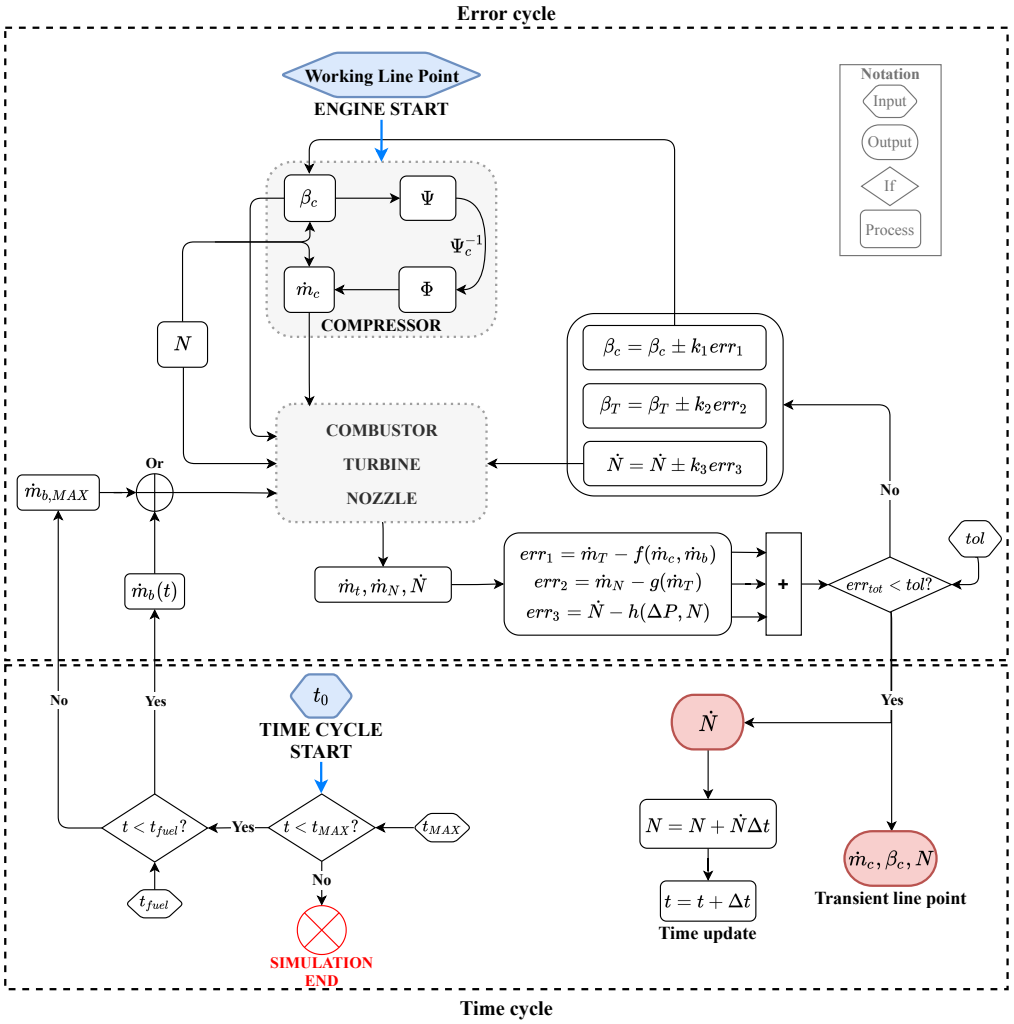


Figure 5.5: *Transient simulation flowchart*

Just like in the previous one, the core of the model is represented by two nested cycles, which are now enclosed into dashed black boxes. The **Error cycle** is devoted to finding the engine parameters that minimise the error produced by the required conditions. As in the steady state simulation we included the mass flow equivalence equations, but the power balance was now replaced by the power excess definition derived from 5.3:

$$error = \dot{N} - \frac{1}{\omega I_P}\left(P_T - \frac{P_c}{\eta_m}\right)$$

The **Time cycle**, on the other hand, is the real update of the transient simulation code, as it describes the integration process mentioned above. The cycle starts from an initial

time value given as an input, and through every iteration until the maximum established time it updates the time-dependent variable, namely the corrected speed N, to determine the relative engine operating parameters.

As a whole, then, the simulation receives as inputs a point over the steady-state working line and a starting time value, and for every following instant determines the new equilibrium point over the compressor map. At the end of the simulation all the output points will create the **transient working line**. In order to conduct a transient analysis we assume, therefore, that the steady-state performance is known from previous stages of the work, so one of the first inputs we give is the complete working line. This is necessary because the starting point of the simulation must be chosen from a steady-state point of the map, otherwise right from the first time-step the engine would try to get back to a stable condition following an unpredictable path.

To determine how severe the transient will be, the simulation is provided (as visible on the left hand of the flowchart) with a further input parameter, the **final fuel injection time** t_{fuel}. Its purpose is to define the instant between t_0 and t_{MAX} when the fuel will stop increasing and remain constant, simulating the moment the throttle is no longer operated. The shorter t_{fuel} is, the quicker the throttle has been hit, meaning that the transient will get the engine to work closer to the surge line.

All these considerations can be evaluated in figure 5.6, where we simulated the transient response of the engine to different throttle inputs.

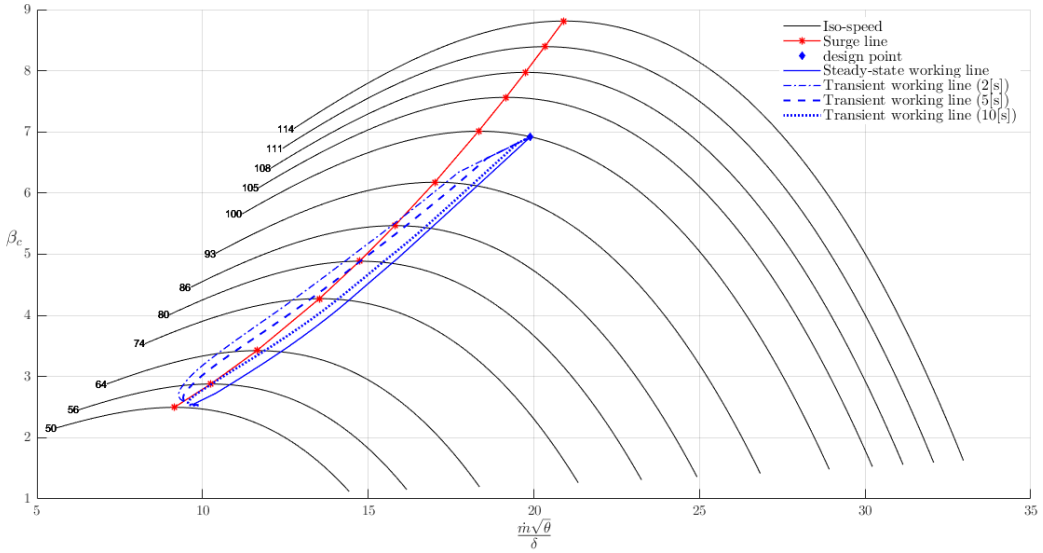
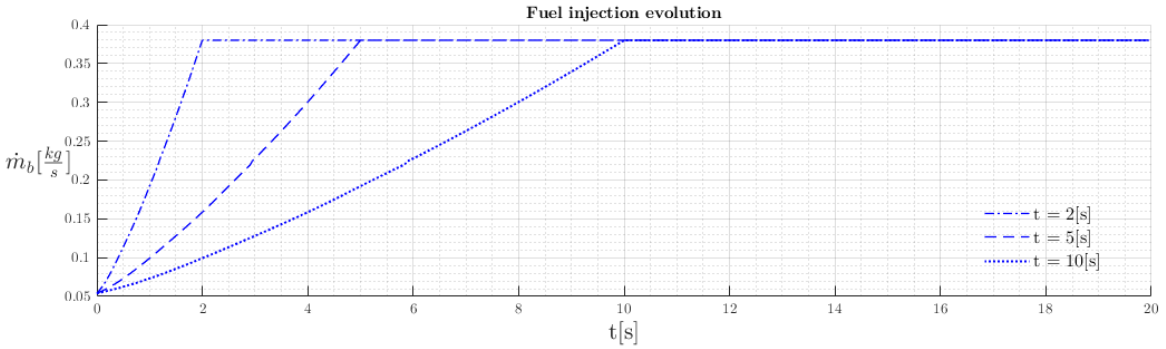
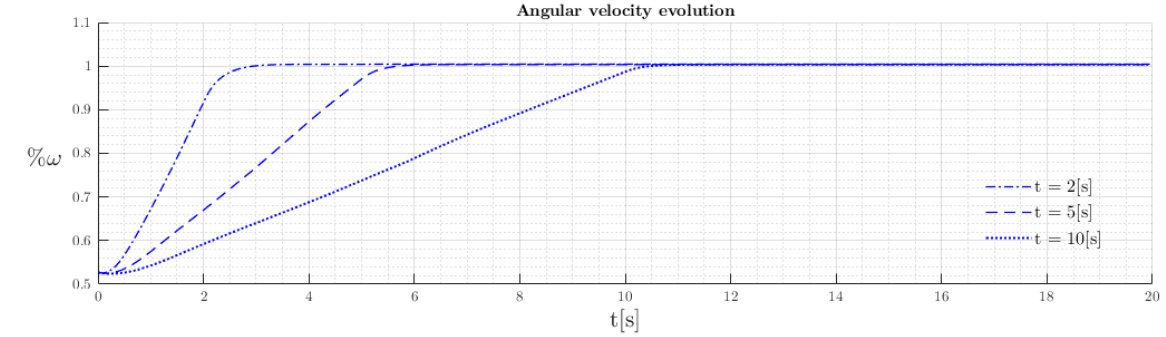
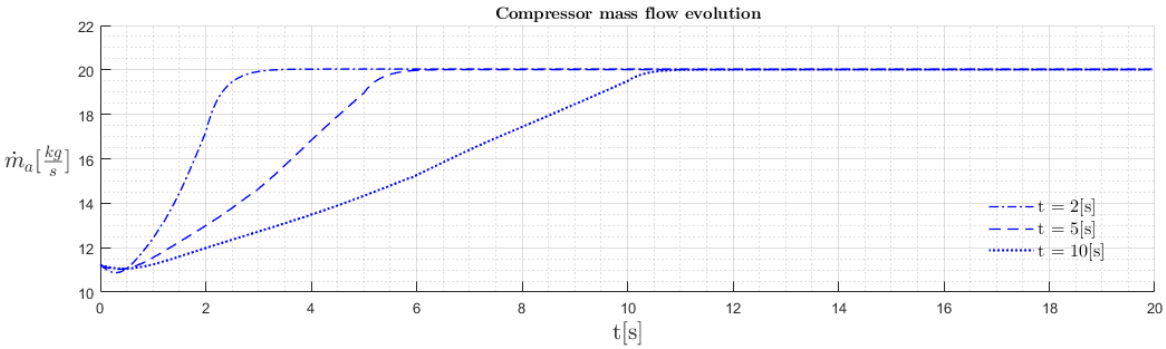


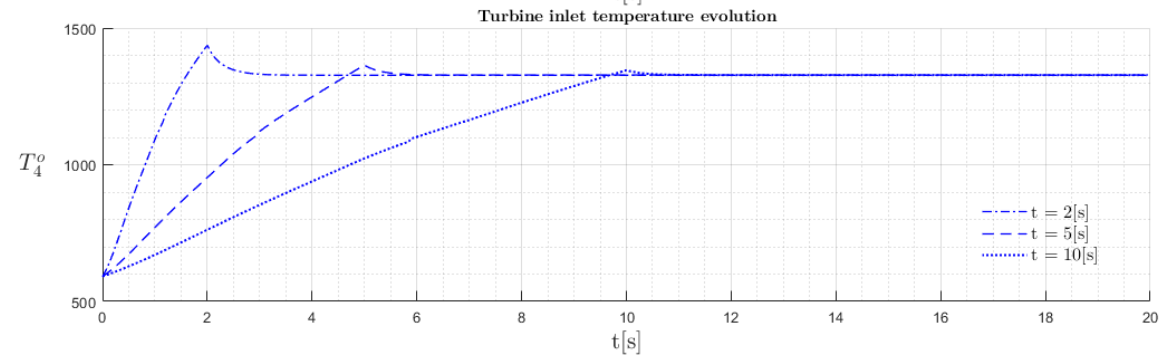
Figure 5.6: *Transient working line under different throttle operations*

The simulations cover all a timespan of 20s, whereas the the fuel increase cut-off was set to three increasing values. It is evident that the 2s transient line describes the most stressful condition for the engine, which is even brought **beyond** the surge line for a broad range of speed values. For a deeper comprehension of the three different situations we report the time evolution of some significant engine parameters.









The first plot shows, as expected, that the fuel is injected in the combustor with a linear increase up until the three chosen t_{fuel} values, and then remains constant till the end of the simulation. This results in similar trends for angular speed ω and compressor mass flow

$\dot{m}_a$. Both of these parameters adapt to the fuel injection with almost the same behaviour, with the exception of a slight time lag which is due to rotating parts inertias. Finally from the T_4° plot we can observe one possible outcome of a hard transient: the temperatures tend to overshoot the stationary values with peaks of increasing height that may determine dangerous problems in the turbine.

Chapter 6

A predictive control using the Moore-Greitzer model

So far we focused our attention on introducing all the tools that were necessary to set the basis for our main work, that is simulating compressor instabilities through an extended engine model. First, we gave a thorough description on how the experimental data was treated in the development of a new, approximated model of the axial compressor. Then, we outlined the basic features of the complete engine simulation, particularly dwelling on how that model was integrated in the environment.

Now that all the groundwork has been exhausted, we can finally get to the core of the project, and go on combining the Moore-Greitzer theory with our one-spool turbojet simulation. This integration was carried out in a **predictive control** approach, where the instability model was used to foresee, and therefore avoid, abnormal conditions.

Before getting into this final simulation, however, a simple **stability analysis** of our engine system will be conducted, and an example of proportional control will be presented.

6.1 Stability analysis

In the theoretical introduction presented in chapter 2 we have already examined the two most common types of instabilities that occur during compressor operation, namely **surge** and **rotating stall**. In order to understand how and why those dangerous behaviours emerge, we first need to perform a **stability analysis** of our dynamical system. According to stability theory, an equilibrium point can be classified in two major categories:

- **stable**, whether the system reacts to a sudden perturbation by getting back to the equilibrium;
- **unstable**, whether the system reacts to a sudden perturbation by amplifying it, and therefore slipping away from the equilibrium.

In this thesis, we applied **linear stability theory** to determine the nature of every point on our compressor characteristic, so that we knew what to expect when solving Greitzer's model. Linear stability studies the system's Jacobian matrix to recognise the different equilibrium points by analysing the **eigenvalues** λ_i. In particular, four types of result are observed:

1. $\lambda_i < 0, \in \mathbb{R}$ corresponds to a stable point;
2. $\lambda_i > 0, \in \mathbb{R}$ corresponds to an unstable point;
3. $\lambda_i \in \mathbb{C}$ with $\Re(\lambda_i) < 0$ corresponds to a stable point with damped oscillations;
4. $\lambda_i \in \mathbb{C}$ with $\Re(\lambda_i) > 0$ corresponds to an unstable point with amplified oscillations.

For a more complete overview of stability analysis of axial compressors refer to [13] and [14].

By performing linear stability analysis to the compressor constitutive equations we assigned every possible operating point to one of these categories, and the results are shown in figure 6.1.

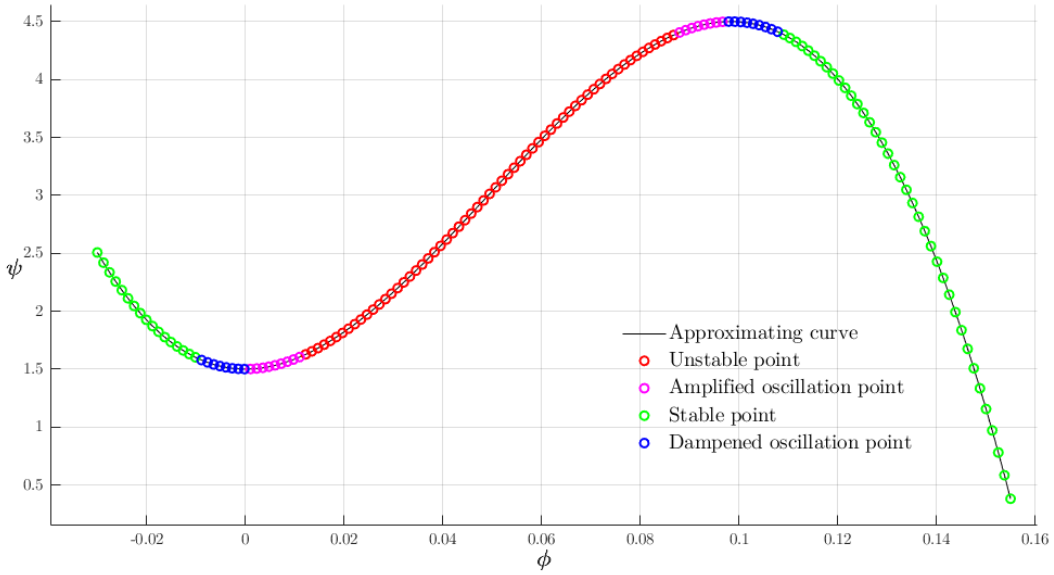


Figure 6.1: *Equilibrium points classification of compressor characteristic*

The results are the confirmation of all the assumptions we made in the previous chapters. Moving backwards from the region on the right of the maximum we stable, so any perturbation does not induce the system to escape from the operating point. The closer we get to the maximum, though, the longer the systems takes before settling on the definitive value, and oscillations begin to show up. On the other hand, if we start operating on the increasing branch of the curve, the system begins slipping away from the equilibrium, first by amplified oscillations, and then exponentially diverging away from it. Finally, when we get past the cubic curve minimum the system is again stable, with properties that are symmetrical to the other decreasing branch.

The real importance of this analysis emerges when we apply these results to the $\beta_c - \dot{m}_c$ map. Since there is a direct correspondence between that and the compressor characteristic, we can assume that the behaviour of the latter can be projected on the former. Considering what we assessed so far, then, not all the points on the right of the surge line can be assumed to be stable. On the contrary, we can delimit a region where oscillations - despite being dampened -, can deeply destabilise the engine, as represented in the red coloured area of figure 6.2. Therefore when we chose the engine design point on the map, or when we evaluate

the position of the working line, we must keep in mind that non-stationary phenomena can occur also in an apparently safe position.

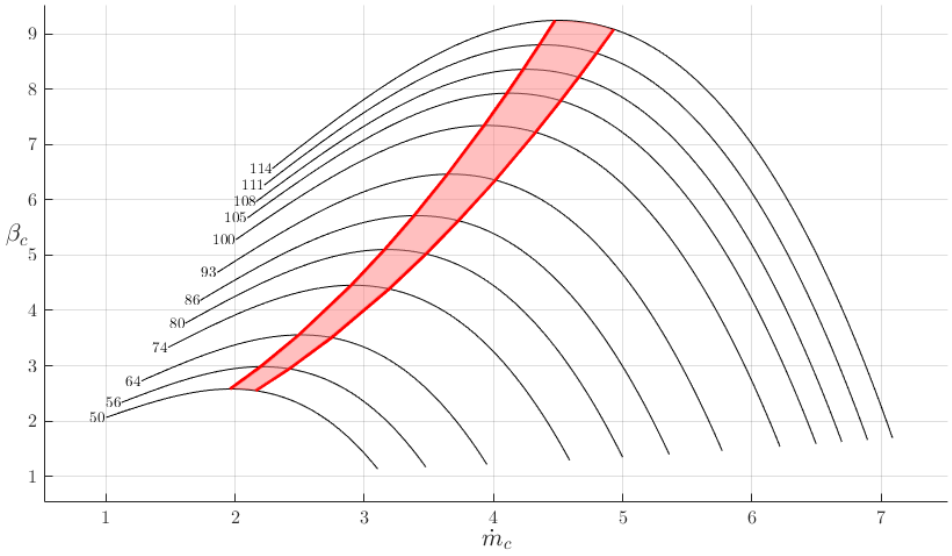


Figure 6.2: *Oscillating region of the compressor map*

6.2 Proportional control on transient performance

To avoid that the transient working line, even under the most stressful operations, does not surpass the surge line, it is possible to introduce in the simulation a **controller**, which is an algorithm that intervenes if a certain condition is not met.

We will report here an example of a **proportional control** applied to the fuel flow value, used to moderate the throttle demand if it becomes too dangerous. The idea is that, if the fuel injection is so quick that the system would be required to operate in the unsafe region of the map, the controller reduces the $\dot{m}_b$ value and forces below the surge line.

Usually, this problem is dealt with by defining a more conservative threshold than the surge line, called **minimum control line**, which sets a limit to the iso-speed lines even before they reach their maximums. The position of this new line can be freely set with the purpose of avoiding the oscillating region represented in 6.2. In figure 6.3, for example, we chose to locate the limit at a relative distance of 8% from the surge line.

The introduction of a minimum control line allows us to replace our former definition of **surge margin** with a new parameter, the **minimum control margin** SM_0:

$$SM_0 = \left(\frac{\beta_{c,MCL}}{\beta_{c,WL}} - 1 \right) \times 100. \tag{6.1}$$

The proportional control intervenes whenever a point of the transient line passes the minimum control line (SM_0 becomes negative). As previously mentioned, the controlled variable is the fuel flow rate, which is altered according to the following law:

$$\dot{m}_{b0} = \dot{m}_b(1 + k \min(SM_0, 0)) \tag{6.2}$$

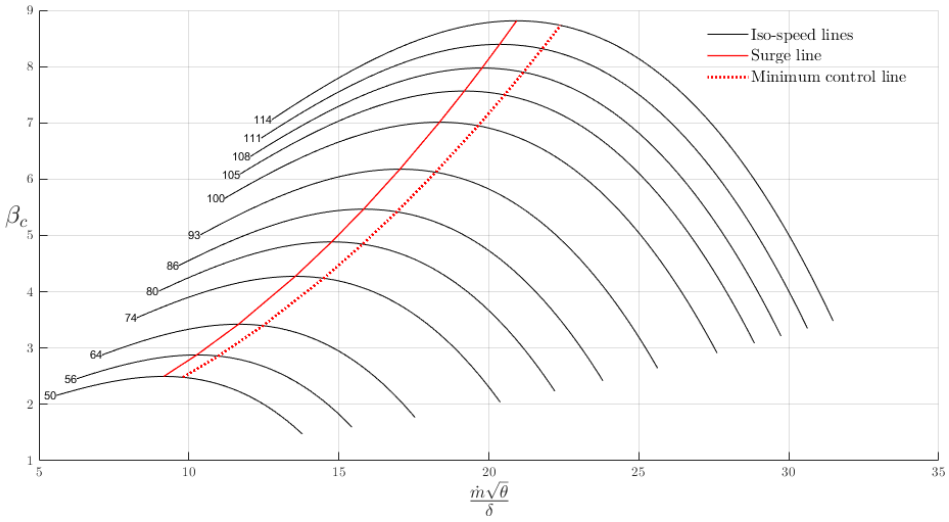


Figure 6.3: *Minimum control line on compressor map*

Basically, when $SM_0 > 0$ $\dot{m}_b$ isn't changed, otherwise it is reduced proportionally through the *control gain* k.

We report here an example of proportional control of the transient performance.

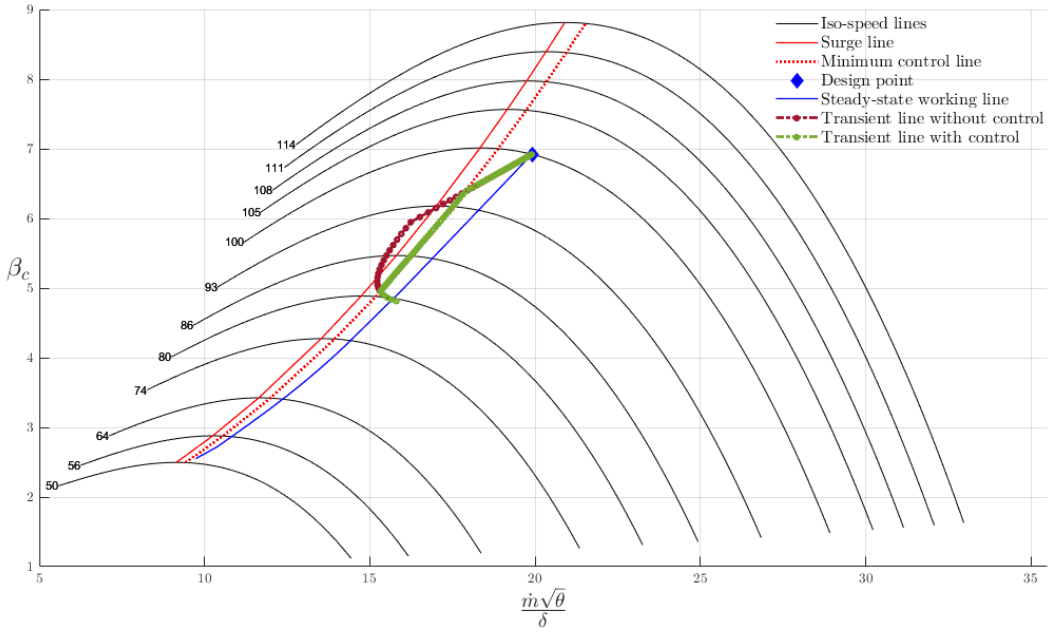
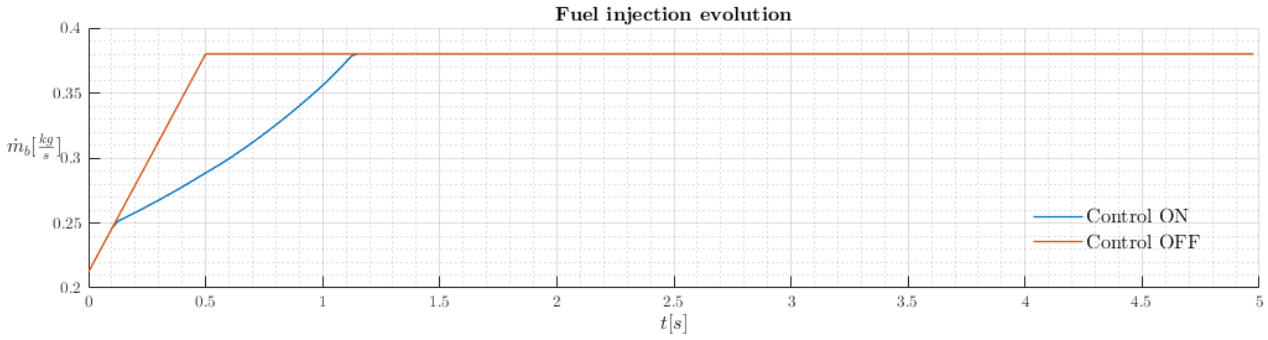


Figure 6.4: *Proportional control of the transient line*

The simulation starts from the point of the steady-state working line corresponding to $N[\%] = 80\%$, and reacts to the usual ramp fuel injection. It is clear that, as far as the engine operates below the minimum control line, the two transient lines coincides. The moment

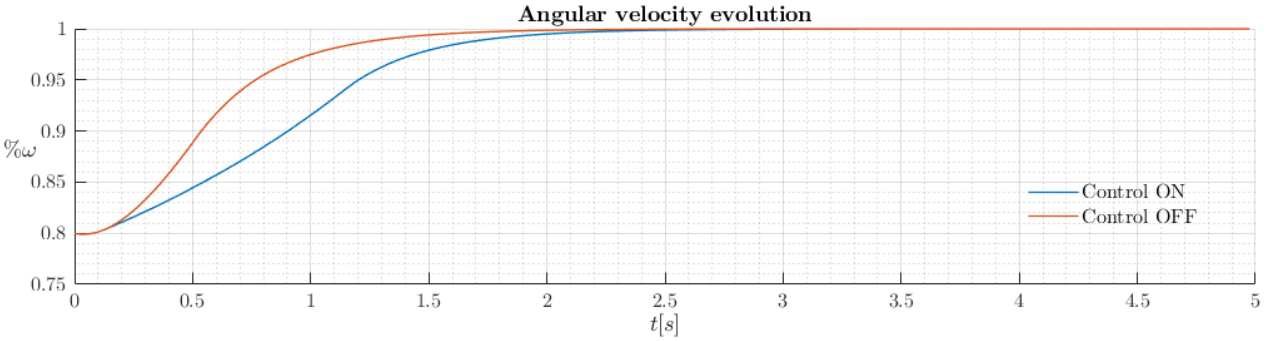
the system would get past that threshold, however, we see that the controlled transient line (green) departs from the other, and moves along the limit without crossing it.

The effect of the control is clearer in the fuel evolution plot.



The blue line shows that the fuel schedule is heavily moderated when the control is active, and therefore the system requires more time before getting to the stationary condition. As a matter of fact, the transient is slowed down by the controller, so all the system variables change less brusquely.

An example of this is represented below.



6.3 Engine simulation with instability prediction

Using Greitzer's model numerical solutions described in chapter 3, we finally have all the ingredients to develop the central part of this thesis: an extended engine model that can foresee the occurrence of unstable behaviours. To pursue this result, we needed to bring together two fundamental components:

1. a complete yet practical engine simulation, to lay out the core environment for the analysis;

2. a reliable mathematical model that could reproduce and predict non-stationary phenomena in the axial compressor.

So far, we thoroughly outlined them both, so it is now possible to proceed with the implementation of one into the other.

Theoretically speaking, introducing Moore and Greitzer's model into the simulation corresponds to adding new time-dependant variables into the system, so we expect to see the effects especially on the **transient response** results.

To describe this complete simulation we will not treat, as in chapter 5, *On Design* and *Off Design* performances, since they were not touched by our latest modifications. On the other hand, Transient analysis will be the object of our attention.

If we observe the results showed in figure 5.6, we may see - as already mentioned -, that a sufficiently quick throttle law can easily bring the system over the surge line and into the dangerous realm of the map. In the last paragraph we described a quite brutal way of avoiding this risk during operation, based on Surge Margin. That is just one in the wide and abundant variety of control systems used in the design of a turbomachine, which either prevent unsafe behaviours or cut them off the moment they occur.

Our purpose here is to develop a new, predictive way of controlling the engine to make sure that these deviations from a safe path are avoided. In particular, we are interested in adopting the Moore-Greitzer model throughout the whole simulation in a **Model Based Predictive Control** logic. The idea is that, for every operating condition computed by the main transient analysis, the current variables will be introduced into a Greitzer simulation to foresee if after a determined period of time the system will still be stable. In case the model detects a risk of instability is coming, a control on the fuel injection will be deployed to moderate the throttle.

Of course, such peculiar integration of engine model and instability prediction is made possible by using the correct compressor map. Since the same nondimensional characteristic is adopted in both simulations, we are sure that the results are coherent and reliable.

Furthermore, the advantage of this logic is that the danger is detected with significant forewarning, therefore the control law does not need to be too complex or invasive: a simple **proportional control** is adopted. We applied a similar approach on the Surge Margin in the previous chapter, with the difference that in that case the limit threshold was fixed for every operating point on the map. Now we are going to define a new controlled variable that is time dependant, and that therefore can be very different depending on the situations.

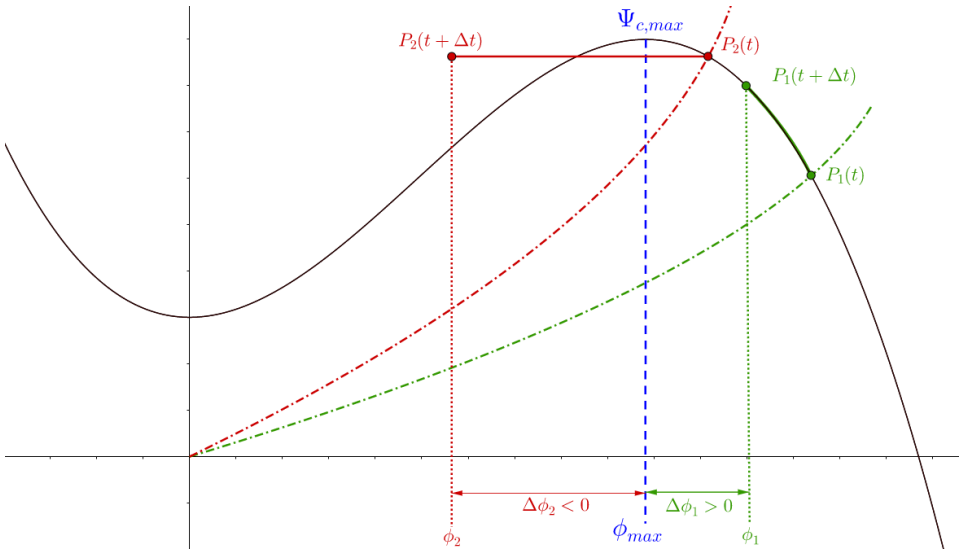


Figure 6.5: *Proportional control logic*

In support of our explanation, we refer to figure 6.7.

If we consider the nondimensional compressor characteristic, we may distinguish two regions based on stability:

- the part where $\phi > \phi_{max}$, which is located on the right of the maximum value of the curve;
- the part where $\phi < \phi_{max}$, located on the left on the maximum.

As demonstrated in the first paragraph of this chapter, our system is stable as long as it operates on the decreasing branch of the curve, and possibly outside the oscillating zone. Keeping this into consideration, then, we choose ϕ_{max} as a threshold variable for our proportional control, defining a new **Predicted Safety Margin** as the horizontal distance $\Delta\phi = \phi - \phi_{max}$ (in percentage) from that value:

$$PSM = \left(\frac{\Delta\phi}{\phi_{max}} \right) \cdot 100$$

The Predicted Safety Margin needs to be positive for the system to be stable, so any time it becomes negative we are falling into surge or rotating stall.

Looking at the figure above, we see two different situations where the control can be applied. Let's assume that the engine is brought to operate at a point corresponding to P_1 on the characteristic, which is represented in green. Through Greitzer's simulation, we evaluate that after an interval Δt the system has moved up on the curve to a point closer to the maximum, but still at an abscissa ϕ_1 such that $\Delta\phi_1 > 0$. This condition determines a positive safety margin, so the system is still **stable**.

Let's now consider point P_2 and its relative throttle line, which are coloured in red. P_2 at instant t is already in a risky region of the curve, so we suppose that during a timespan Δt a surge cycle has started that moved it away from the stationary characteristic, towards $\phi = \phi_2$. It is clear that this new operating point is well beyond the stability threshold, and in fact $\Delta\phi_2 < 0$: the system is **unstable**.

For every input coordinates given by the engine simulation, then, we will compute the value of PSM to determine whether the system is moving towards a stable condition or not, and if a risk of instability is detected, a proportional control steps in.

It is important to mention that the use of ϕ_{max} as threshold is a very permissive choice in terms of control. If a stricter condition is necessary, we may easily set the limit on a larger flow coefficient value, so that instability is prevented even before the curve maximum is passed.

In figure 6.6 is reported the updated transient simulation flowchart that integrates the control logic in the system. Error and time cycle blocks are left implicit, but the general layout is similar to the original one (5.5). For every time-step input parameters go through the cascade of engine components, the output values are evaluated to compute residual errors, and based on those errors the initial parameters are updated until convergence is reached.

The real addition resides, of course, in the **control block** located between the compressor and the other engine components. Its purpose is to receive from the compressor the current global variables, and determine the fuel flow rate that keeps the system stable.

This is done with the introduction of a **predictive control modified** $\dot{m}_b$, which exploits a proportional logic to moderate the fuel injection, based on the PSM value computed by our **Greitzer Solver**.

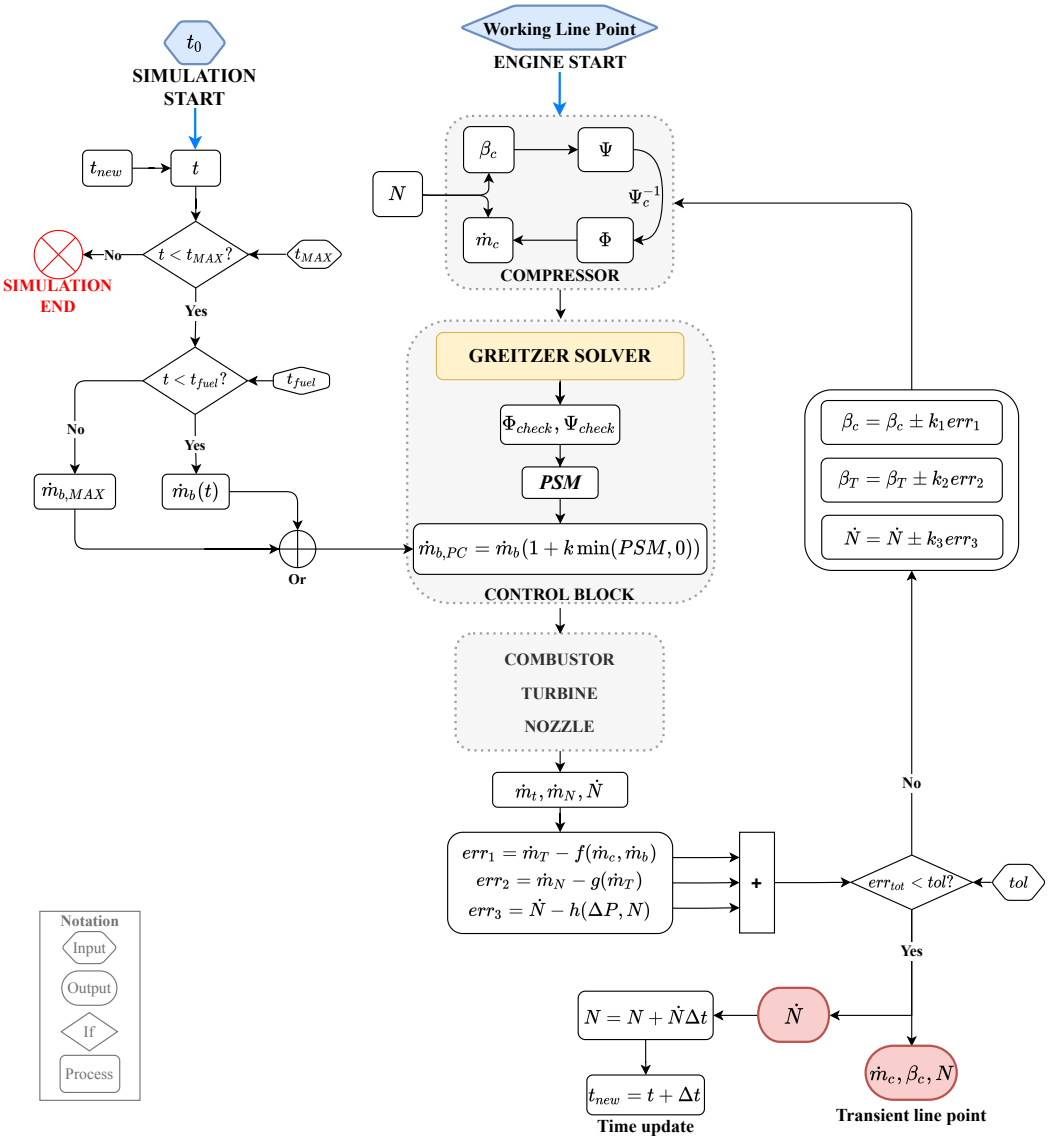


Figure 6.6: *Transient simulation with predictive control flowchart*

Its definition is the following:

$$\dot{m}_{b,PC} = \dot{m}_b(1 + k\min(PSM, 0)) \tag{6.3}$$

Constant k is the **proportional gain**, chosen with **positive** sign and used to determine the correction magnitude on the system variable.

To understand better how the control block operates we developed a specific flowchart, showed below.

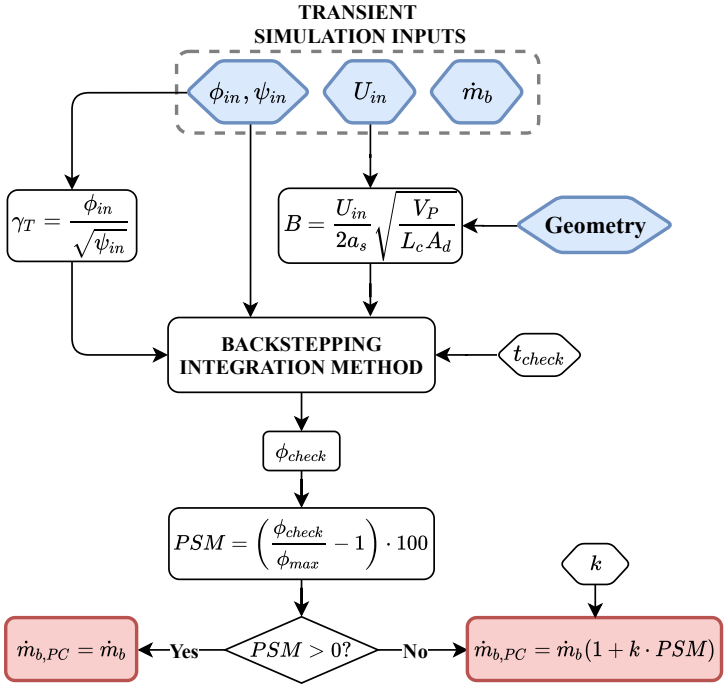


Figure 6.7: *Control block flowchart*

The controller receives from the engine four parameters:

1. a **flow coefficient** ϕ, which is derived from the current mass flow rate;
2. a **pressure coefficient** ψ, derived from the nondimensionalisation of the current pressure ratio;
3. a **tangential velocity** at mean radius, computed from current corrected speed;
4. the current **fuel flow rate** imposed by the fuel schedule.

The first three values are necessary for the application of our Greitzer model integration method, described in the previous section of this chapter. More specifically, a starting point for the integration is established on the compressor characteristic in terms of ϕ_{in} and ψ_{in}. These two values are used also in the definition of the **throttle gain**, to guarantee that the operating point of the Moore-Greitzer model corresponds to that of the engine. The tangential velocity, then, is introduced in the parameter B definition, which needs to be coherent with the current operating point.

An important parameter that has to be defined before the actual integration process is t_{check}, which corresponds to the time when the integration stops (we previously called this t_{end}) and the variables are controlled. This value is particularly important because it sets the *time-horizon* of the controller: the larger it is, the deeper potential instabilities can get.

Now that all the necessary parameters have been defined, the variables can be integrated through the model we outlined in figure 3.3, which will compute the flow coefficient value after the timespan we established.

The last part of the block contains the control process we described in the last pages, and operates on the $\dot{m}_b$ input. The Predicted Safety Margin is computed with the final value ϕ_{check}, and its sign determines the two possible outcomes of the control:

- $PSM > 0$ means that the system is safe, so the fuel injection is unaltered;
- $PSM > 0$ means that an instability has been detected, so the fuel has to be reduced proportionally.

Based on either of these results the control block returns the parameter $\dot{m}_{b,PC}$, which will then be employed in the engine simulation.

6.4 Real cases application

In this final part of the chapter we are going to see our Model Based Predictive Control applied to a few real scenarios, using the extended simulation code we described so far.

First, the control will be applied to a quite conservative situation to see whether the results seem reliable; then, a more demanding throttle law will be adopted to test our controller in a riskier simulation.

6.4.1 First case: conservative scenario

Our first engine simulation with predicted control has been carried out using a steady-state working line which remained quite distant from the unstable zone of the map. Our purpose was to observe if the prediction model behaved accurately even away from the surge line, in a region where deep instabilities were not expected.

The results from Off Design analysis is shown in figure 6.8

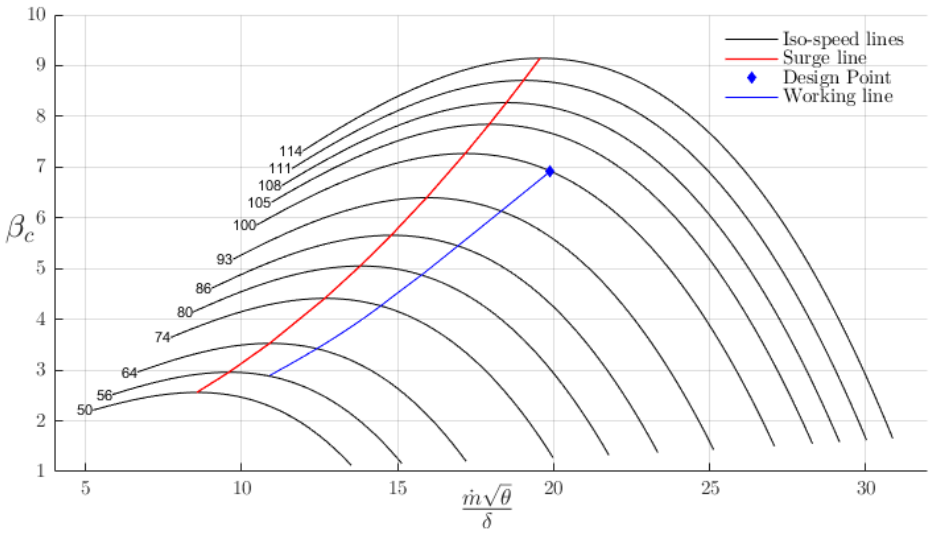


Figure 6.8: *Safe working line configuration*

Design Point was not changed from the original one ($\dot{m}_c = 19.90, \beta_c = 6.92$); what we altered is the scaling factor of the map: a safer operative point was selected on the compressor map.

To conduct the Transient simulation, the following parameters were chosen:

Parameter	*Value*
Engine Simulation	
$N_{in}[\%]$	74
$\dot{m}_{c,in}$	14.50
$\beta_{c,in}$	4.28
$\dot{m}_{b,in}$	0.18
$\dot{m}_{b,max}$	0.38
t_{fuel}	1.5
Controller	
ϕ_{th}	$1.15\phi_{max}$
t_{final}	3
k	0.01

The initial point was set on the steady-state working line in correspondence to $N[\%] = 74\%$, and the usual ramp law was imposed to the fuel schedule, so that the engine should reach nominal value after 1.5 s. As for the controller, a time-horizon of 3 s was set, with time-steps of hundredths of a second. To guarantee that the controller would at some point intervene, we had to establish a very conservative value for our threshold flow coefficient ϕ_{th}, which in fact was set 15% larger than the usual ϕ_{max}.

The results of the simulation are shown in figure 6.9:

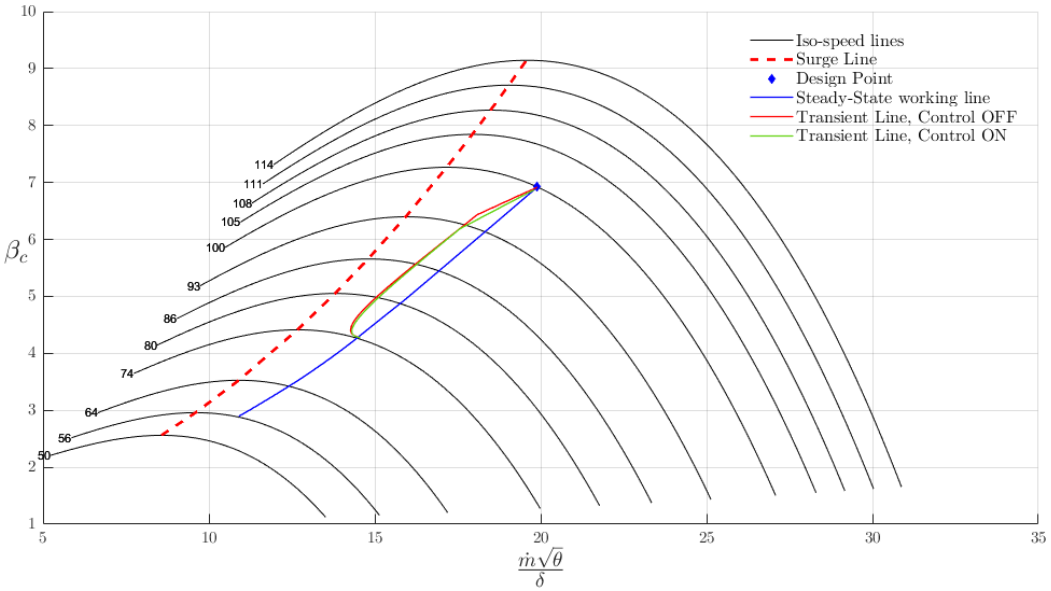


Figure 6.9: *Transient simulation with conservative control*

The red line refers to the transient line **without control**, whereas green line represents the **controlled** transient.

We may see that due to the fact that all steady-state points reside in a very stable region of the map, the controller does not need to be invasive. However, if we get a closer look at

both lines (figure 6.10) we appreciate that the control distinctly moderates the transient, keeping it closer to the steady-state line and away from surge.

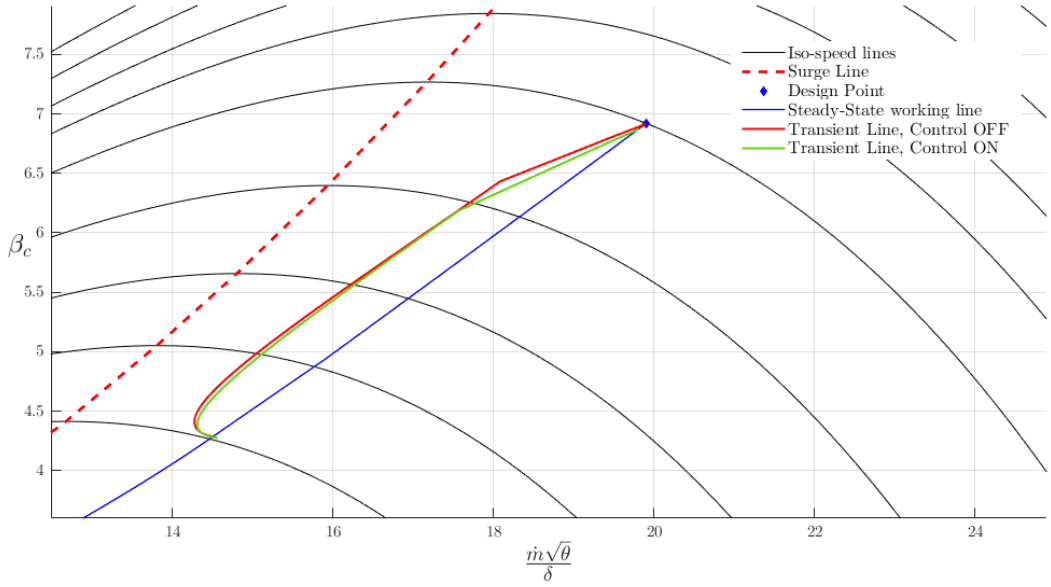


Figure 6.10: *Closer view of control action*

Particularly helpful to observe the controller actions is the time evolution of the fuel flow and the rotational speed, because they show how the system reacts in both circumstances.

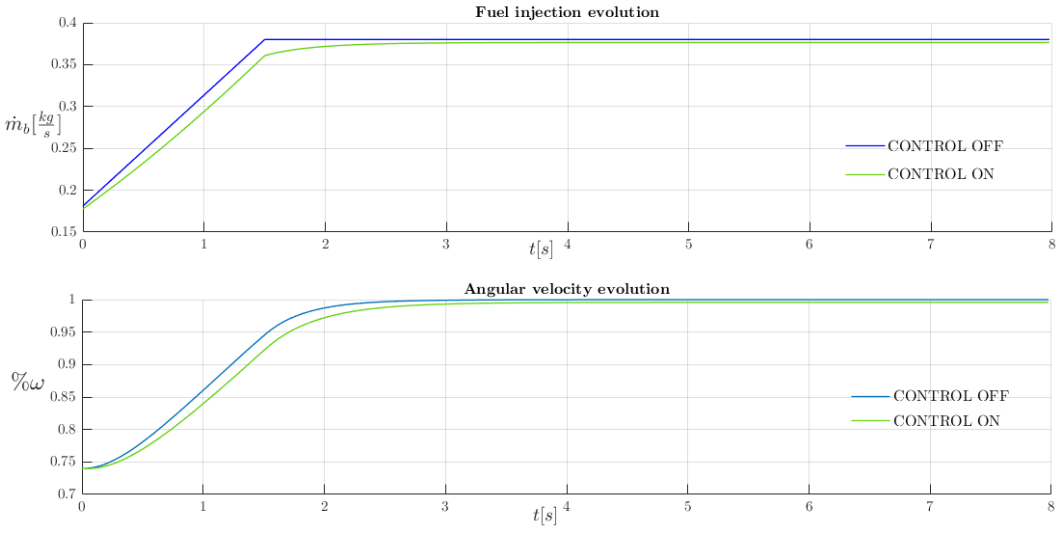


Figure 6.11: *Engine parameters with and without control*

In the first plot above we see that, compared to the nominal ramp input given by the fuel flow injection (blue line), the controlled fuel flow (green) is visibly smaller, and tend

to grow more moderately. This behaviour is reflected on the angular velocity, which reacts similarly.

6.4.2 Second case: stressful configuration

Our second case study involved a more demanding scenario for the engine, in that the steady-state working line moved closer to the surge line. This means that a quick enough throttle law could actually bring the operative point in an unstable region of the map, where dangerous behaviour can occur.

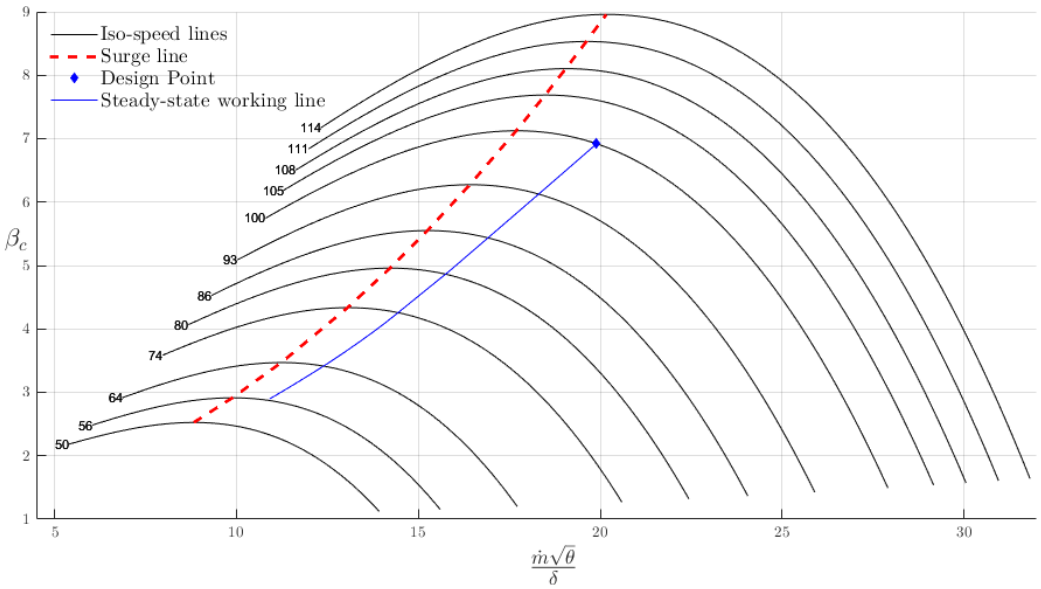


Figure 6.12: *Transient simulation with aggressive control*

The parameters adopted for this simulation were:

Parameter	*Value*
Engine Simulation	
$N_{in}[\%]$	80
$\dot{m}_{c,in}$	15.71
$\beta_{c,in}$	4.84
$\dot{m}_{b,in}$	0.22
$\dot{m}_{b,max}$	0.38
t_{fuel}	1.3
Controller	
ϕ_{th}	$1.1\phi_{max}$
t_{final}	3
k	0.03

This time a quicker throttle law has been imposed to further stress the engine, whereas the control threshold has been set at 10% the maximum value.

As before, we can now observe the results of the simulation over the compressor map.

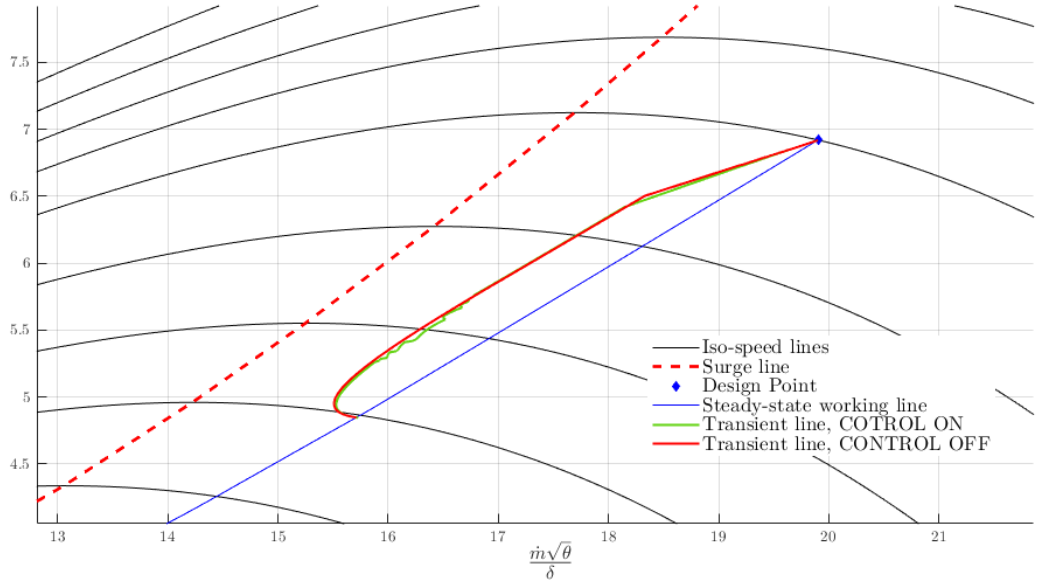


Figure 6.13: *Control action over an aggressive working line*

Once again, the controlled transient line lies below the free system line, as a symptom of the controller action. The alterations are not very evident, and are mostly limited to the initial part of the line, which is also the most stressful for the engine. We can see that in that segment points are not linearly aligned, but follow the scattered geometry imposed by the control operation. In the final part, instead, where the engine moves farther from the surge line, the results of the controlled and un-controlled systems coincide.

Below are reported the engine variables associated to this configuration.

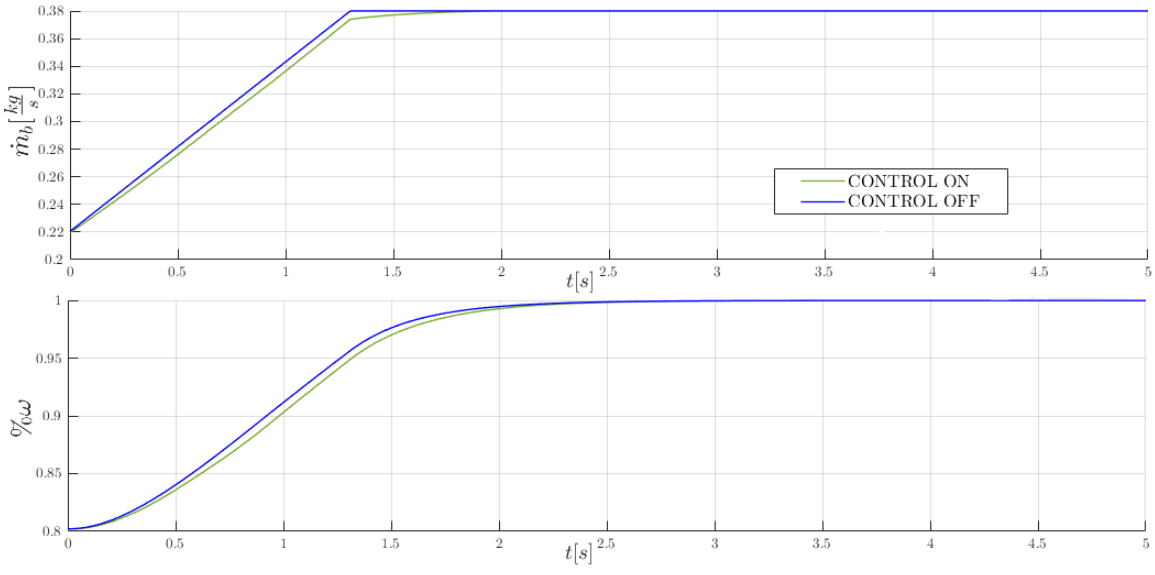


Figure 6.14: *Engine parameters with and without control*

The fuel flow appears again moderated by the controller, which distributes the injection process over a longer timespan.

If this particular simulation behaved very efficiently, with results much similar to what we expect from a traditional proportional control, something particular happens the moment we speed up the throttle law. Keeping the other parameters unaltered, we slightly decreased the final fuel injection time to the value of $t_{fuel} = 1\,\mathrm{s}$. We expect from this change that the un-controlled systems evolves with a more demanding dynamic, with a transient line getting closer to the surge line. From figure 6.15 below we see that that was actually what happened, with the controller forced to intervene deeply to contain the transient response.

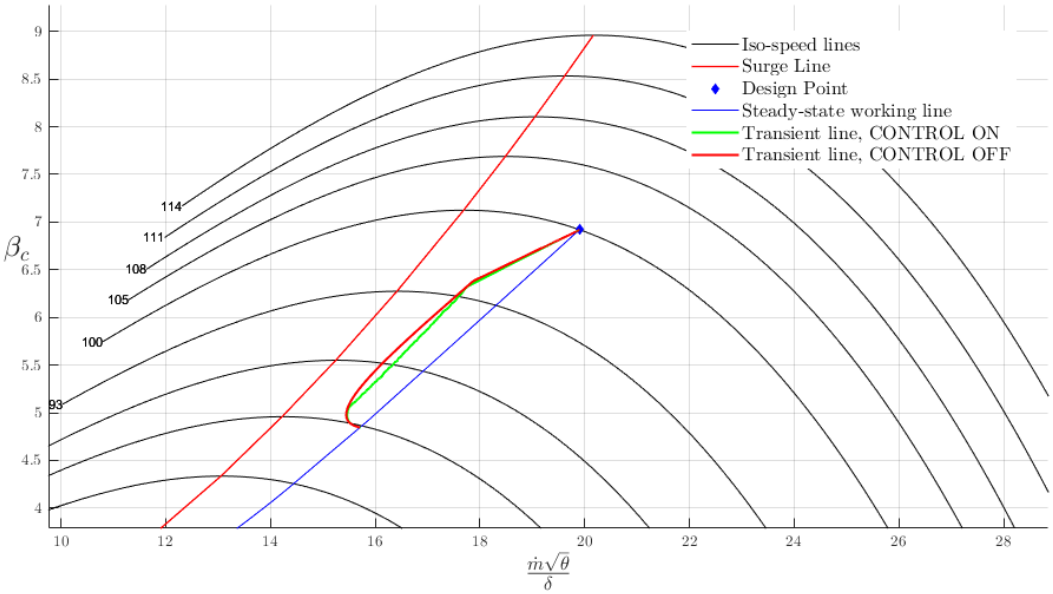


Figure 6.15: *Transient response to quick throttle law*

The real peculiarity of this configuration emerges when we analyse the time evolution of our usual variables, as reported below in figure 6.16.

When we observe the effect of the controller on the fuel ramp input, we see that there are specific time values where the fuel flow instantaneously shrinks into negative peaks. As a whole, these sporadic deviations do not influence the general transient response of the engine, and in fact for parameters like the angular velocity they are simply averaged out (second plot of the figure).

This means that they are directly derived from the control logic, which affects specifically the value of $\dot{m}_b$. Observing the output values for every instant of the simulation, we realised that the peaks are not random results, but instead they are correctly determined by the integration of the Moore and Greitzer model. As a matter of fact, some particular engine conditions, if projected over the same time-horizon used for all iterations, result in much more unstable behaviour, and even **deep surge**.

When this happens, the model computes a much higher value of the PSM, and therefore the correction over the nominal fuel flow becomes particularly invasive. An example of one of these anomalies has been reproduced through a Greitzer integration simulation to verify what we explained so far, and the results confirmed the hypothesis.

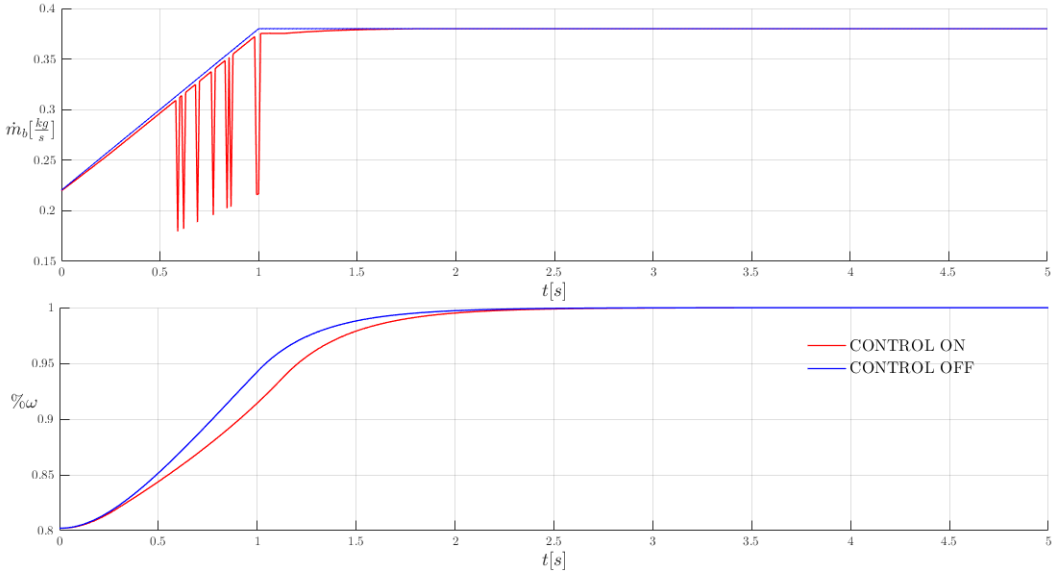


Figure 6.16: *Anomalies during controlled simulation*

In figure 6.17 we clearly see that, after the three second integration required by the control system, a surge cycle has begun that brings the output flow coefficient well beyond the safety threshold.

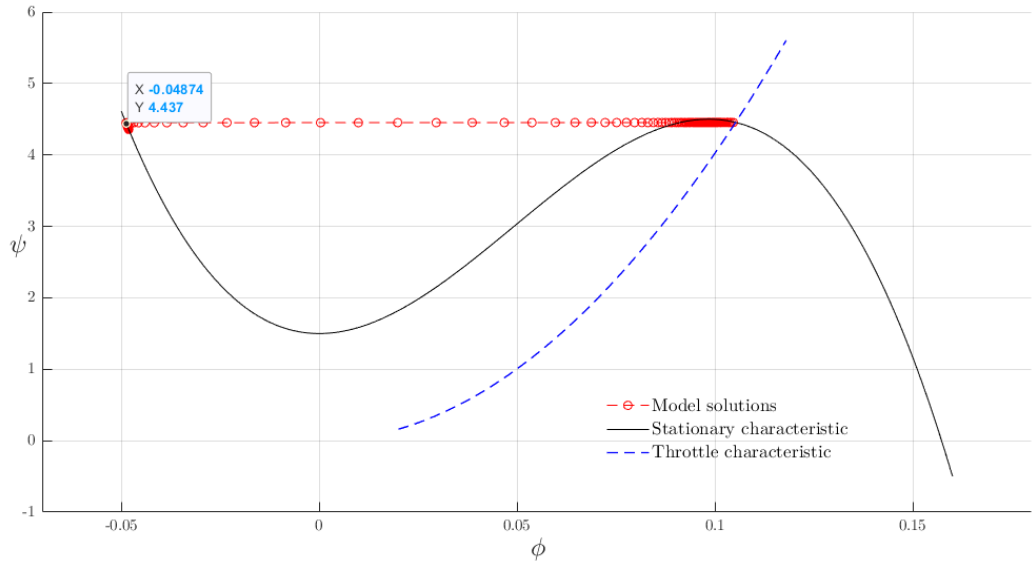


Figure 6.17: *Deep surge insurgence captured by the predictive controller*

Since an outcome like this can only be associated to the presence of surge, a possible correction to the controller can be added to reduce its effect. More specifically, making use of the fact that surge determines a null value of J variable, a simple modification of the

proportional control allows to change the magnitude of the control gain if surge is detected.

An example is the following:

$$\dot{m}_{b,PC} = \dot{m}_b(1 + \alpha_k k \min(PSM, 0)),$$

where to α is assigned a very small value if the current $J \simeq 0$, else it is simply kept equal to unity.

Bibliography

[1] E. M. Greitzer. Surge and Rotating Stall in Axial Flow Compressors - Part I and II. *Journal of Engineering for Gas Turbines and Power*, April 1976.

[2] F. K. Moore and E. M. Greitzer. A Theory of Post-Stall Transients in Axial Compression Systems: Part I - Development of Equations. *Journal of Engineering for Gas Turbines and Power*, January 1986.

[3] F. K. Moore. A Theory of Rotating Stall of Multistage Axial Compressors: Part II - Finite Disturbances. *Journal of Engineering for Gas Turbines and Power*, April 1984.

[4] F. K. Moore. A Theory of Rotating Stall of Multistage Axial Compressors: Part III - Limit Cycles. *Journal of Engineering for Gas Turbines and Power*, April 1984.

[5] National Aerospace Laboratory NLR. **GSP 11**, `Gas Turbine Simulation Program`, 2009.

[6] C. Drummond and C.R. Davison. Improved Compressor Maps Using Approximate Solution to the Moore Greitzer Model. *ASME Turbo Expo Technical Conference*, June 2009.

[7] W. A. Tesch and W. G. Steenken. Blade Row Dynamic Digital Compression Program Volume I, J85 Clean Inlet Flow and Parallel Compressor Models. *NASA CR-134978*, March 1976.

[8] R. D. Hager. Analysis of Internal Flow of J85-13 Multistage Compressor. *NASA TM X-3513*, April 1977.

[9] V. Timo. A numerical tool simulating unsteady dynamics of a microturboshaft engine for UAV applications. *MSc. thesis*, Politecnico di Torino, 2015.

[10] A. F. Spesot. Gas turbine engine performances in sub-idle conditions. *MSc. thesis*, Politecnico di Torino, 2016.

[11] J. D. Mattingly, W. H. Heiser, D. T. Pratt. Aircraft engine design, second edition. *AIAA Educational series*, 2002.

[12] J. Kurzke and I. Halliwell. Propulsion and Power - An Exploration of Gas Turbine Performance Modeling; chapter 7. *Springer*, 2018.

[13] Alberto Campagna. Dinamica di instabilita' fluidodinamiche in compressori assiali. *Tesi di Laurea*, Politecnico di Torino, 2017.

[14] C. A. Mansoux, D. Gysling, J. D. Setiawan and P. James . Distributed nonlinear modeling and stability analysis of axial compressor stall and surge. *American Control Conference*, 1994.