

# Modern Gain Scheduling

## Overview

This chapter offers a relatively short overview of alternative gain scheduling methods that have been proposed the last years. These methods are the LPV, velocity-based and neural/fuzzy gain scheduling approaches and they play also a significant role on the subject, being extensively used on real-world systems. Since these methods have not been given further consideration in this work, the material referenced here is not exhaustive and is presented as a bibliographic complement to Chapter 1 detailing linearization-based gain scheduling techniques.

## Chapter contents

---

<b>2.1</b>	<b>LPV Gain Scheduling . . . . .</b>	<b>49</b>
2.1.1	Polytopic Approach . . . . .	49
2.1.2	LFT Approach . . . . .	51
<b>2.2</b>	<b>Velocity-based Gain Scheduling . . . . .</b>	<b>53</b>
<b>2.3</b>	<b>Neural/Fuzzy Gain Scheduling . . . . .</b>	<b>55</b>

---

## 2.1 LPV Gain Scheduling

The LPV (Linear Parameter Varying) gain scheduling approach is the major alternative to the linearization-based one presented in Chapter 1. This method has some advantages over the latter since it offers more serious stability guaranties for the gain-scheduled system. Even though it may also be used for purely LPV plants resulting from linearization of a nonlinear parameter-dependent system, it is most interesting when it is applied to an over-bounding  $q$ -LPV reformulation of the nonlinear system. This method can also incorporate bounds on the scheduling vector rates (and thus reducing conservatism) using parameter-dependent Lyapunov functions. However in some cases it may be rather conservative due to this reformulation incorporating redundant trajectories that may not belong to the nonlinear system and in addition, it does not offer feasibility guaranties for the existence of the gain-scheduled controller.

Motivation

During the last fifteen years there has been a true wealth in the bibliography on these methods and several issues continue to be treated; the following analysis attempts to give only some of the basic results concerning this approach, being mainly divided in two major categories: polytopic and LFT gain scheduling.

### 2.1.1 Polytopic Approach

The polytopic approach in LPV gain scheduling is very common in the scientific community and has been extensively studied; the results presented here are taken mainly from [13]. The main goal behind this approach is to calculate a gain-scheduled controller  $K(s, \varrho)$  that will guarantee internal stability and in addition quadratic  $\mathcal{H}_\infty$  performance  $\gamma$  on the performance vector  $\zeta_\infty$  over external disturbances  $w$  and for all admissible values of the scheduling vector components  $\varrho_i$  (see Fig. 2.1).

Control goal

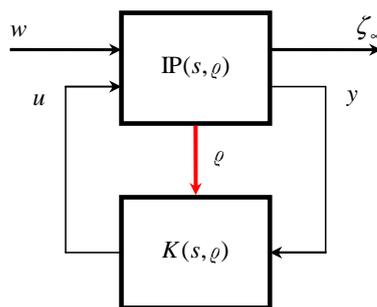


Figure 2.1: Polytopic gain-scheduling structure.

The class of systems considered have an affine dependence of their state space matrices on the scheduling vector components and in addition the scheduling vector takes values inside a convex polytope<sup>1</sup>.

<sup>1</sup>As a result the state-space matrices take also values inside a convex polytope.

Searching for a (single) Lyapunov matrix  $\mathbf{X} = \mathbf{X}^T > 0$  that satisfies the BRL<sup>2</sup> ensuring quadratic  $\mathcal{H}_\infty$  performance for a generic (not polytopic) LPV (or  $q$ -LPV) plant poses an *infinite* number of constraints; however for the polytopic case, the problem is tractable and reduces to a *finite* number of constraints posed for each vertex of the polytope. This is the result of the famous *vertex property* stating that the following two arguments are equivalent:

Vertex  
property

- The polytopic LPV system is stable with quadratic  $\mathcal{H}_\infty$  performance  $\gamma$ .
- There exist a single matrix  $\mathbf{X} = \mathbf{X}^T > 0$  satisfying the collection of LMI's<sup>3</sup>:

$$\mathcal{B}_{[\mathbf{A}_{cl}^i, \mathbf{B}_{cl}^i, \mathbf{C}_{cl}^i, \mathbf{D}_{cl}^i]}(\mathbf{X}, \gamma) < 0, \quad i = 1, \dots, r.$$

Controller  
computation

The gain-scheduled LPV controller sought will also be of polytopic form and the LPV synthesis problem is primarily to find a common Lyapunov matrix for all vertices; this is done considering the corresponding set of the classic LMI feasibility conditions, given also in Section 3.3.3.2 of this report. The LMI's are in fact solved for two matrices  $\mathbf{R}, \mathbf{S}$  and the Lyapunov matrix is finally constructed solving some matrix equations.

Once the feasibility conditions are met and the Lyapunov matrix  $\mathbf{X}$  computed, all vertex controllers:

$$\mathbf{\Omega}_i = \begin{pmatrix} \mathbf{A}_k^i & \mathbf{B}_k^i \\ \mathbf{C}_k^i & \mathbf{D}_k^i \end{pmatrix} \quad (2.1)$$

may be sequentially computed either by solving the BRL's either by the same convex optimization algorithms or symbolically. The final LPV controller will be of the form

$$K(\varrho) : \begin{aligned} \dot{x}_k &= \mathbf{A}_k(\varrho)x_k + \mathbf{B}_k(\varrho)y \\ u &= \mathbf{C}_k(\varrho)x_k + \mathbf{D}_k(\varrho)y \end{aligned} \quad (2.2)$$

and its matrices computed as a convex combination of the vertex controllers, using the current/measured value of the scheduling vector  $\varrho(t)$ .

Additional  
work

For a similar treatment of the problem see [18]; being one of the first works on the subject. A multi-objective approach treating simultaneously  $\mathcal{H}_\infty$  and  $\mathcal{H}_2$  performances, passivity, asymptotic disturbance rejection, time-domain constraints and constraints on the closed loop location for different channels of the closed loop system with a common Lyapunov function is given in [115]. A good reference on advanced gain-scheduling techniques in order to reduce computational burden is proposed in [9]. Parameter-dependent controllers if the scheduling parameters are real are using a skew-symmetric technique is presented in [117] whereas some work using the elimination lemma is proposed in [126]. Finally, other more recent approaches may be found in [5, 127, 154].

<sup>2</sup>Bounded Real Lemma.

<sup>3</sup>The symbol ' $\mathcal{B}$ ' denotes BRL and  $\mathbf{A}_{cl}^i, \mathbf{B}_{cl}^i, \mathbf{C}_{cl}^i, \mathbf{D}_{cl}^i$  are the closed loop standard model matrices at each of the  $r$  vertices of the polytope (see also [13], Eq. 29).

### 2.1.2 LFT Approach

The LFT (Linear Fractional Transformation) gain scheduling approach is an important alternative to the standard Lyapunov-based LPV one. Perhaps the one of the first and most widely known attempts to deal with this formulation can be found in [12] and the material presented in this section is mainly drawn from there.

This method is different from the classical polytopic one since it does not require an affine dependence of the system's state space matrices on the time-varying parameters  $\varrho_i$  since they enter both the plant & controller dynamics in a specific way. In addition, concerning implementation, the parameter vector need not be expressed as the convex combination of its vertex values. However, this method may become cumbersome when the plant's LFT form needs to be computed since it is highly non-unique and the designer may need repeated parameter blocks, thus increasing the order of the system.

LFT vs.  
Polytopic

The main idea behind this method is to re-cast the initial gain scheduling problem as one of robust performance in the face of structured uncertainty using small gain theory. In the beginning the designer performs the following procedure: starting from a generic nonlinear parameter-dependent plant  $\mathcal{S}_{\text{pd}}$  as in Eq. 1.22 (with  $\varrho(t)$  being the on-line measured parameter/scheduling vector) obtains a  $q$ -LPV re-formulated system  $\mathcal{S}_{q\text{-LPV}}$  (see Eq. 1.24) or an LPV one using Jacobian linearization (see Eqs. 1.26-1.29). From this point, an equivalent u-LFT formulation of this system is calculated as a specific connection of a block-diagonal parameter block  $\Theta$ , containing the measured parameters (see Eqs. 1.30-1.31 and Fig. 1.5)<sup>4</sup>. This u-LFT connection is written as:

Modeling

$$\begin{bmatrix} \zeta_{\infty} \\ y \end{bmatrix} = \mathcal{F}_u(\mathbb{P}, \Theta) \begin{bmatrix} w \\ u \end{bmatrix} \quad (2.3)$$

where  $\zeta, y, w, u$  are once again the performance, controller input, perturbation and controller output vectors respectively.

The gain-scheduling problem now is to find a time-varying controller having a similar l-LFT formulation as the plant; thus find a control input  $u$  that satisfies:

Control  
goal

$$u = \mathcal{F}_l(K, \Theta). \quad (2.4)$$

This specific structure of the 'time-varying' uncertainty interconnection between the system, the controller and the scheduling vector block containing the varying parameters is depicted in Fig. 2.1. Alternatively, the overall feedback connection may be expressed as

$$T(\mathbb{P}, K, \Theta) = \mathcal{F}_l(\mathcal{F}_u(\mathbb{P}, \Theta), \mathcal{F}_l(K, \Theta)) \quad (2.5)$$

and the corresponding  $\mathcal{H}_{\infty}$  control problem may now be posed as follows:

<sup>4</sup>The parameters are supposed to be confined to a ball with radius  $\gamma^{-1}$ , with  $\gamma$  being the corresponding  $\mathcal{H}_{\infty}$  performance level; however a re-scaling on the input perturbations may be performed in order to permit larger variations.

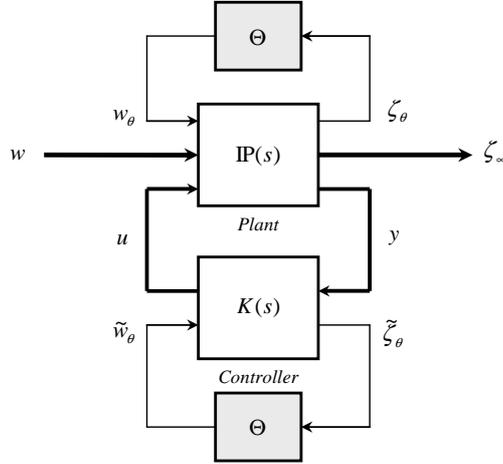


Figure 2.2: LFT gain-scheduling structure.

Find a structure (i.e. state space matrices) for the controller  $K(s)$  such that the gain-scheduled LPV controller  $\mathcal{F}_l(K, \Theta)$  satisfies the following properties:

- The closed loop system given by Eq. 2.5 is internally stable for all parameter trajectories  $\varrho_i(t)$  that satisfy the scaling  $\gamma^2 \Theta^T \Theta \leq 1$ .
- The induced  $\mathcal{L}_2$  norm of the closed loop system satisfies:

$$\max_{\|\Theta\|_\infty \leq \gamma^{-1}} \|T(\mathbb{P}, K, \Theta)\|_\infty \leq \gamma.$$

Now the closed loop system may be written as the u-LFT of the l-LFT of a specific augmented system  $\mathbb{P}_{aug}(s)$  (containing  $\mathbb{P}(s)$ ) with the unknown controller  $K(s)$ , and a  $2 \times 2$  block diagonal matrix containing all measurable time-varying parameters:

$$T(\mathbb{P}, K, \Theta) = \mathcal{F}_u \left( \mathcal{F}_l(\mathbb{P}_{aug}, K), \begin{pmatrix} \Theta & 0 \\ 0 & \Theta \end{pmatrix} \right). \quad (2.6)$$

Controller  
computation

From the aforementioned equation it may be deduced that the original gain scheduling problem may be viewed as a classical robust performance problem in the face of the block-repeated uncertainty and sufficient conditions for solvability are provided by the small gain theory. This results to a particular case of the general  $\mathcal{H}_\infty$  synthesis problem which is rather easily transformed to a set of LMI existence conditions that may be solved using interior point algorithms. Finally, the controller matrices are computed by the corresponding to the  $\mathcal{H}_\infty$  problem scaled BRL (see also Section 3.1.3).

Globally, a nice introduction to the subject may be found in the survey papers [88] and [114], whereas more modern results concerning this method are given in [14, 67, 94, 148, 151, 153].

## 2.2 Velocity-based Gain Scheduling

The velocity-based gain scheduling method belongs to the class of methods characterized by the ‘divide and conquer’ design presented in Chapter 1. It has been already remarked that (see Section 1.3.2.2) gain-scheduled controller realizations ensuring correct trim control and matching of the linearization of the gain-scheduled controller, at each equilibrium/synthesis point, with the corresponding member of the controller family designed at this point, may be designed.

This property is often called *local linear equivalence* (see [88], §3.1b) but it does not treat the extent for which this equivalence is valid. A particular class of methods claiming to treat this subject was developed in the late 90’s by D. J. Leith and W. E. Leithead with a series of articles and is called velocity-based implementation of the gain scheduling controllers, satisfying an *extended* local linear equivalence property. Given the fact that this method has not received the appropriate attention henceforth and due to the already mentioned exchange in [84], its potential remains yet to be proved.

Motivation

This method differentiates itself from the classical first order series (or Jacobian) linearization theory used by every method mentioned so far, in order to obtain a family of linear systems computed at a corresponding family of equilibrium points of a nonlinear parameter-dependent system. The most important point here is that when considering linearized models such as the ones in Eq. 1.26, the notation  $x_\delta = x - x_{\text{eq}}$  is abused since this difference quantity only *tends* to describe the true state difference and only under heavy assumptions on the corresponding linear system’s range of operation, on the input rate etc.

In addition, when a nonlinear system is not confined to a vicinity of an equilibrium point, the linear approximation of Eq. 1.26 does not offer an accurate approximation of the nonlinear system dynamics. To overcome this difficulty, Leith & Leithead offer an alternative linearization around a generic operating point  $(x_1, u_1)$  of the system (see [85], §3.2, 3.3). The linearized model thus is<sup>5</sup>:

Modeling

$$\begin{aligned}\dot{\hat{x}}_\delta &= \nabla_x \mathbf{f}(x_1, u_1) \hat{x}_\delta + \nabla_u \mathbf{f}(x_1, u_1) u_\delta + \mathbf{f}(x_1, u_1) \\ \hat{y}_\delta &= \nabla_x \mathbf{h}(x_1, u_1) \hat{x}_\delta + \nabla_u \mathbf{h}(x_1, u_1) u_\delta\end{aligned}\quad (2.7)$$

with  $u_\delta = u - u_1$ ,  $\hat{y}_\delta = \hat{y} - y_1$ ,  $\hat{x}_\delta = \hat{x} - x_1$  and  $\dot{\hat{x}}_\delta = \dot{\hat{x}}$ , with the neighborhoods around the operating point being sufficiently small. The main difference with the Jacobian linearization is that there is a nonzero term in the first equation<sup>6</sup> thus making the approximation model nonlinear. Re-arranging the previous equation using the aforementioned transformations, the following approximation of the system’s state is obtained:

$$\begin{aligned}\dot{\hat{x}} &= \nabla_x \mathbf{f}(x_1, u_1) \hat{x} + \nabla_u \mathbf{f}(x_1, u_1) u + \mathbf{f}(x_1, u_1) - \nabla_x \mathbf{f}(x_1, u_1) x_1 - \nabla_u \mathbf{f}(x_1, u_1) u_1 \\ \hat{y} &= \nabla_x \mathbf{h}(x_1, u_1) \hat{x} + \nabla_u \mathbf{h}(x_1, u_1) u + \mathbf{h}(x_1, u_1) - \nabla_x \mathbf{h}(x_1, u_1) x_1 - \nabla_u \mathbf{h}(x_1, u_1) u_1\end{aligned}\quad (2.8)$$

<sup>5</sup>Dependence on the scheduling vector is for now omitted for simplicity.

<sup>6</sup>Note that in the Jacobian case this term vanishes since this point is an equilibrium one.

**Velocity form** Now by differentiating Eq. 2.8 and considering the appropriate initial conditions for the state of the system, one gets the following *velocity form* that is now totally linear:

$$\begin{aligned}\dot{\hat{x}} &= \hat{w} \\ \dot{\hat{w}} &= \nabla_x \mathbf{f}(x_1, u_1) \hat{w} + \nabla_u \mathbf{f}(x_1, u_1) \dot{u} \\ \dot{\hat{y}} &= \nabla_x \mathbf{h}(x_1, u_1) \hat{w} + \nabla_u \mathbf{h}(x_1, u_1) \dot{u}.\end{aligned}\tag{2.9}$$

From the analysis found in [85], it turns out that the aforementioned linear system yields an approximation of the initial nonlinear system's dynamics during a certain time interval for the operating point considered, accurate now to a *second* order (instead of a first one in the Jacobian case). It is evident that additional linearizations are needed for subsequent operating points, when the approximation error starts to increase.

**Controller computation** Suppose now that the initial nonlinear system is in fact dependent on the scheduling vector with  $\varrho = \varrho(x, u)$ ; then the approximation performed in Eq. 2.9 is now *scheduling vector-dependent*. Based on this modeling, encapsulating an approximation of the nonlinear system for an arbitrary operating point, a gain scheduling procedure may be devised.

This is first done by calculating a family of specific velocity-based linearization type of controllers that achieves the performance requirements for the now linear velocity-based model of the nonlinear plant. Since this linear is smoothly parameterized by the scheduling vector  $\varrho$ , one needs to calculate an infinite number of controllers for every possible value of  $\varrho$ ; however, the same strategy as with conventional gain scheduling may be used where controllers are designed only at a number of operating points.

The final gain-scheduled controller is obtained from the family of linear controllers by permitting the scheduling vector to vary with the operating point. A thorough treatment of the subject is clearly out of the scope of this work but one can refer to the series of papers [82, 83, 86] and additional details may be also found in [87].

**Alternative method** An interesting approach that satisfies however only the local linear equivalence property has been given in [71]. This work once again considers the demand for an appropriate gain-scheduled controller implementation that preserves the input-output properties of the closed loop systems locally about each equilibrium point. The method and the control law proposed follow the so-called  $\mathcal{D}$  procedure and use a particular form in order to construct the controller. Integral action is added at its input whereas some of its inputs are differentiated before actually given to the controller.

It is claimed that this scheme does not introduce any additional noise amplification at the relevant inputs and outputs of the linearized feedback system since all closed loop functions are preserved. However the issue of noise amplification inside the controller and how it impacts on the behavior of the nonlinear feedback system is not addressed.

## 2.3 Neural/Fuzzy Gain Scheduling

An unconventional method to construct gain-scheduled controllers has been developed by the fuzzy/neural community and applied on numerous cases, e.g. for flight control laws; for some recent examples see [70, 107, 141]. The material presented here is mainly from the survey [88] and from the tutorial-like paper [135] reviewing separately classical, fuzzy, neural, and neuro-fuzzy gain-scheduling.

The first step towards the design of a fuzzy gain-scheduled controller is a representation of a nonlinear system as a blend of  $i$  local models of the form: Modeling

$$\dot{x} = \sum_i \bar{\mathbf{f}}_i(x, u) \mu_i(\sigma) \quad (2.10)$$

$$y = \sum_i \bar{\mathbf{h}}_i(x, u) \mu_i(\sigma) \quad (2.11)$$

The functions  $\mu_i$  are the so-called *membership functions* used to blend these models, with  $\sum_i \mu_i(\sigma) = 1$  and the quantity  $\sigma = \sigma(x, u)$  shows the dependence of this blending on the state and the input. The interesting fact in the approach is that the blended models may be considered as affine local models:

$$\bar{\mathbf{f}}_i(x, u) = \alpha_i + \bar{\mathbf{A}}_i x + \bar{\mathbf{B}}_i u \quad (2.12)$$

$$\bar{\mathbf{h}}_i(x, u) = \beta_i + \bar{\mathbf{C}}_i x + \bar{\mathbf{D}}_i u \quad (2.13)$$

This blending representation can thus directly lead to a divide and conquer gain scheduling strategy since a local controller may be designed for a local model and then blended using the weighting functions, according to the quantity  $\sigma$ . This representation of the nonlinear system may be also considered in another context: each member of the local models family may be used only at a certain operating region of the system, leaving the blending occur at transition regions; however, with this approach problems occur concerning coupling terms with the derivatives of the membership functions. Controller computation

The primary advantage of a fuzzy gain-scheduled controller is that the plant modeling may be done exploiting human expertise on particular systems where modeling using the physics laws of physics is not possible or does not lead to reliable results. However, this procedure of determining a fuzzy model may be time consuming and demanding extensive computer simulations to reassure the designer for the closeness of the fuzzy model to the real-world system. Features & comments

This leads to neural network-based gain scheduling that utilizes the learning capabilities of a neural network so that the controller parameters are ‘learned’ without a detailed prior knowledge of the plant. This method also has drawbacks since a neural-network does not give much insight into the plant dynamics and its structure is not an easy to task to construct. Thus, combined schemes may be used that take advantage of each approaches benefits (see [135] and references therein).



## Chapter 3

# Control Theory for Gain Scheduling

### Overview

A major advantage of gain scheduling control is that it provides *nonlinear* parameter-dependent systems a *nonlinear* time varying controller by using *linear* time invariant ones. It has been remarked that for real world applications the elegant and powerful results of the modern  $\mathcal{H}_\infty$  control theory are particularly interesting for the synthesis of LTI controllers in contrast to other methods such as predictive and/or pure nonlinear control strategies that risk being overly complex and/or difficult to implement. In this work two  $\mathcal{H}_\infty$  control structures were tested in order to provide the necessary LTI controllers needed for interpolation in the gain scheduling control context. This chapter offers a solid yet not exhaustive review of two of these methods:  $\mathcal{H}_\infty$  dynamic output feedback with pole placement constraints and  $\mathcal{H}_\infty$  dynamic and static loop shaping. In addition some rather standard results concerning full order state observers and Youla parametrization (in use with the first synthesis method) and an important system analysis tool called the *gap metric* (in use with the second synthesis method) are presented.

## Chapter contents

---

<b>3.1</b>	<b><math>\mathcal{H}_\infty</math> Control in LMI Regions</b>	<b>59</b>
3.1.1	Motivation	59
3.1.2	LMI Regions	64
3.1.2.1	<i>Design Objectives</i>	64
3.1.2.2	<i>D-Stability</i>	65
3.1.3	Controller Synthesis	67
<b>3.2</b>	<b>Compensator Estimator-Controller Form</b>	<b>70</b>
3.2.1	Motivation	70
3.2.2	Controller Transformation	70
<b>3.3</b>	<b><math>\mathcal{H}_\infty</math> Loop Shaping</b>	<b>73</b>
3.3.1	Motivation	73
3.3.2	The Loop Shaping Design Procedure (LSDP)	79
3.3.3	Full Order Case	82
3.3.3.1	<i>Standard Solution</i>	83
3.3.3.2	<i>LMI Solution</i>	84
3.3.4	Static Case	87
<b>3.4</b>	<b>The Gap Metric</b>	<b>89</b>
3.4.1	Motivation & Definitions	90
3.4.2	Connection to the $\mathcal{H}_\infty$ Theory	93
3.4.3	Computation of the Gap Metric	94
<b>3.5</b>	<b>Conclusions</b>	<b>95</b>

---

### 3.1 $\mathcal{H}_\infty$ Control in LMI Regions

In this section some theoretical results concerning  $\mathcal{H}_\infty$  control with pole placement constraints in LMI regions will be presented. The section starts with a classical analysis motivating the use of this powerful synthesis method for the computation of LTI controllers at the first benchmark example of Chapter 5. The subsequent sections give all the necessary results for a systematic treatment of this control problem with most of the material drawn from [27].

#### 3.1.1 Motivation

Consider a SISO linear time invariant system  $G(s)$  and a controller  $K(s)$  in a standard closed loop control configuration (see Fig. 3.1a). The primary goal of classical control systems is to design the controller  $K$  so that the time response  $y(t)$  to a step reference input  $y_r(t)$  has *good* properties. Many of these properties are dominated mostly by the location of the poles  $\lambda$  of the closed loop system  $H(s)$  with:

$$H(s) \triangleq \frac{Y(s)}{Y_r(s)} = \frac{G(s)K(s)}{1 + G(s)K(s)} \quad (3.1)$$

To quantify the influence of the pole location to the time response of the closed loop system  $H(s)$ , suppose that  $H$  is or may be approximated by a second order system (see Fig 3.1b), as is the case very often in practice, with:

$$H(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (3.2)$$

The transfer function parameters  $\omega_n$  and  $\xi$  are called *undamped natural frequency* and *damping ratio* of the poles  $\lambda_{1,2}$  of  $H$ , being the roots of its denominator, with:

$$\lambda_{1,2} = -\xi\omega_n \pm j\omega_n\sqrt{1 - \xi^2}. \quad (3.3) \quad \text{Poles}$$

The quantitative meaning of the two fundamental variables  $\omega_n$  and  $\xi$  is related to the step response of  $H$ . The undamped natural frequency is the system's output oscillation frequency if its damping ratio is reduced to zero whereas the damping ratio is closely related to the overshoot experienced on the system's step response, given that the system is underdamped.

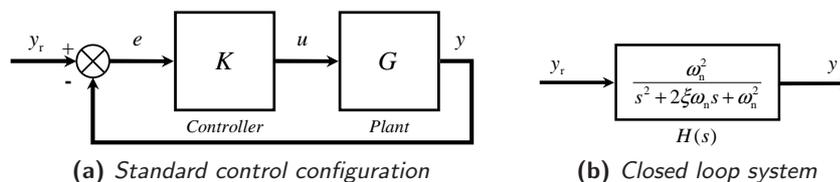


Figure 3.1: Basic analysis block diagrams.

2<sup>nd</sup> order  
system  
analysis

## Damping scenarios

The poles  $\lambda_{1,2}$  depend on both  $\omega_n$  and  $\xi$  (see Eq. 3.3), but it is the latter that characterizes the form of the step response  $y(t)$ . Four scenarios are considered for the damping ratio:  $\xi = 0$  (non-damped),  $0 < \xi < 1$  (underdamped),  $\xi = 1$  (critically damped) and  $\xi > 1$  (overdamped). The first and the third scenarios may be considered as limit cases of the second and the fourth ones.

In the non-damped case ( $\xi = 0$ ) the closed loop poles are purely imaginary with  $\lambda_{1,2}^{\text{nd}} = \pm j\omega_n$  and the time response is purely oscillatory whereas in the critically damped case ( $\xi = 1$ ) the closed loop poles are purely real and negative with equal values  $\lambda_{1,2}^{\text{cd}} = -\omega_n$ . The system in the first case is said to be *conditionally stable* whereas in the second remains always *stable*. In the overdamped case ( $\xi > 1$ ) the system demonstrates two distinct stable real poles with  $\lambda_{1,2}^{\text{od}} = (-\xi \pm \sqrt{\xi^2 - 1})\omega_n$ . For a constant undamped natural frequency, as the damping ratio increases the first stable pole goes to infinity whereas the second goes to zero. Thus, the time response of such a system becomes sluggish since it gets dominated by a slow stable eigenvalue. All three cases are not interesting for a control system for stability and/or speed reasons, so only the underdamped case is considered in the following analysis.

For an underdamped system  $0 < \xi < 1$  its step response  $y(t)$  and step tracking error  $e(t) = y_r(t) - y(t)$  (see Fig. 3.1b) are computed using basic knowledge of ODE theory as (see [105], pp. 147-148)<sup>1</sup>:

$$y(t) = 1 - \frac{e^{-\xi\omega_n t}}{\sqrt{1 - \xi^2}} \sin\left(\omega_d t + \arctan \frac{\sqrt{1 - \xi^2}}{\xi}\right) \quad (3.4)$$

$$e(t) = e^{-\xi\omega_n t} \left( \cos \omega_d t + \frac{\xi}{\sqrt{1 - \xi^2}} \sin \omega_d t \right). \quad (3.5)$$

The step response  $y(t)$  of  $H(s)$  for a given  $\omega_n$ , presents different amounts of overshoot and oscillation around the desired reference trajectory  $y_r(t)$  for different values of  $\xi$  (see Fig. 3.2a) whereas its settling speed for a given  $\xi$  is a function of  $\omega_n$  (see Fig. 3.2b).

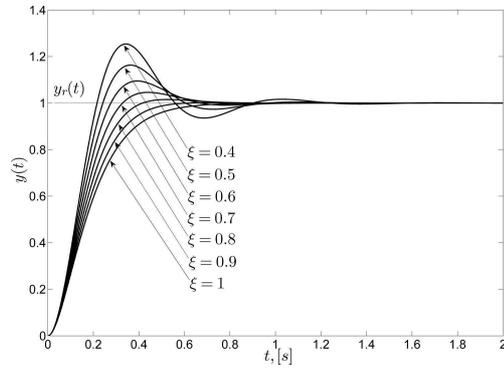
In order to characterize an LTI system in a more uniform way, several properties of its step time response  $y(t)$  may be defined, depending only on the damping ratio  $\xi$  and undamped natural frequency  $\omega_n$ . Some of these properties are the *rise time*  $t_r$ , *peak time*  $t_p$ , *settling time*  $t_s$  and *overshoot*  $M_p$  (see Fig. 3.2c) and may be easily calculated for a second order system as:

## Step response properties

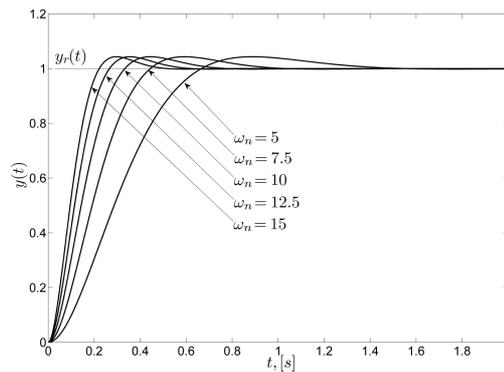
1. *Rise Time*  $t_r$ : It is usually defined as the time that the step response  $y(t)$  takes to reach its 100% value for the first time. It may be computed from Eq. 3.4 by letting  $y(t) = 1$ :

$$t_r = \frac{\pi - \arctan \frac{\sqrt{1 - \xi^2}}{\xi}}{\omega_n \sqrt{1 - \xi^2}}. \quad (3.6)$$

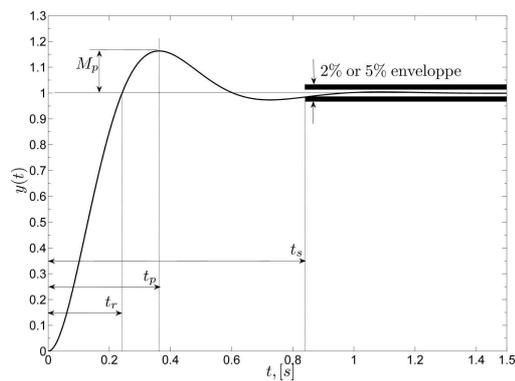
<sup>1</sup>The quantity  $\omega_d = \omega_n \sqrt{1 - \xi^2}$  is called the *damped* natural frequency.



(a) Step responses (varying  $\xi$ )



(b) Step responses (varying  $\omega_n$ )



(c) Step response characteristics

Figure 3.2: Step response study - underdamped case.

2. *Peak Time*  $t_p$ : It is defined as the time that the step response  $y(t)$  takes to reach its maximum value. It is computed by letting the derivative of  $y(t)$  go to zero:

$$t_p = \frac{\pi}{\omega_n \sqrt{1 - \xi^2}} = \frac{\pi}{\omega_d}. \quad (3.7)$$

3. *Settling Time*  $t_s$ : It is defined as the time that the step response  $y(t)$  takes to reach a 2% or 5% envelope around its steady state value  $y(t_\infty)$ . It is approximatively computed as:

$$t_s = \frac{3}{\xi \omega_n} \quad (5\% \text{ criterion}) \quad (3.8)$$

$$t_s = \frac{4}{\xi \omega_n} \quad (2\% \text{ criterion}) \quad (3.9)$$

4. *Maximum Overshoot*  $M_p$ : It is defined as the maximum positive percentage deviation (occurring at the peak time  $t = t_p$ ) of the step response  $y(t)$ . It is computed as:

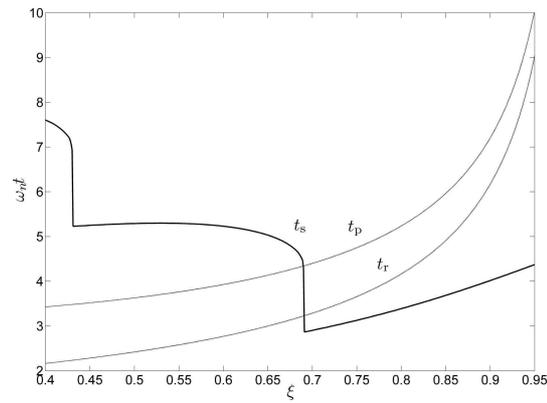
$$M_p = \frac{y(t_p) - y(t_\infty)}{y(t_\infty)} \cdot 100\% = e^{\frac{-\pi\xi}{\sqrt{1 - \xi^2}}} \cdot 100\% \quad (3.10)$$

Pole  
placement  
discussion

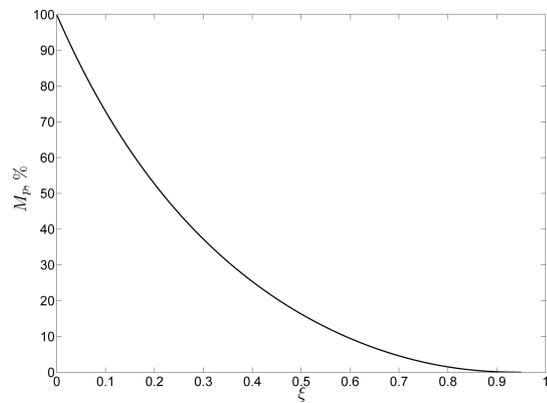
A control system should be able to provide satisfactory *response times* and *damping* for the plant under control. For a second order system with the simple form of Eq. 3.2, this is done by placing its poles  $\lambda_{1,2}$  (see Eq. 3.3) to an appropriate location following two rules of thumb, as it has been implied in the preceding analysis: first the desired settling time  $t_s$  of the process is set by adjusting the undamped natural frequency  $\omega_n$  and then an appropriate damping ratio  $\xi$  is chosen in order on the one hand avoid excessive overshoot, and on the other hand obtain a time response for the system that is not too sluggish.

The dependence of the rise, peak and settling times over the damping ratio and a given undamped natural frequency is shown in Fig. 3.3a<sup>2</sup>. Even though the rise and peak times augment monotonically with the damping ratio  $\xi$ , it does not happen the same with the settling time. It may be remarked that while the settling time is almost constant for medium values of the damping ratio  $0.45 \leq \xi \leq 0.65$ , it reaches a minimum for  $\xi \simeq 0.69$  and then starts to rise almost linearly. The corresponding percentage overshoot  $M_p$  for this optimal value of the damping ratio is about 4.7% (see Fig. 3.3b). In practice, a damping ratio between 0.6 and 0.8 for the closed poles of a real-world system is considered satisfactory with the undamped natural frequency being chosen as a function of the specific bandwidth demanded from the control system.

<sup>2</sup>The figure shows the settling time for  $\omega_n = 1 \text{ rad/s}$  and thus provides scaling for any  $\omega_n$ .



(a) Step response times



(b) Step response overshoot

**Figure 3.3:** Step response characteristics.

### 3.1.2 LMI Regions

Motivation  
for  
eigenvalue  
clustering

From the analysis of the previous section it has been made clear that the transient behavior of a control system is dominated by the location of its closed loop poles. For a simple second order system as the one in Eq. 3.2, it is generally easy to obtain the desired closed loop dynamics by setting the damping ratio and undamped natural frequency to some desired values. For a higher order system there exist also solid methods for robust state/output feedback eigenvalue placement to an arbitrary accuracy (see for example [72] for details on the algorithm implemented in MATLAB<sup>®</sup> for state feedback eigenvalue placement).

Besides focusing on eigenvalue placement only, a control system could provide a control law that takes into account constraints over frequency domain aspects, robustness over external perturbations and parametric uncertainties. A good way to take into account all these requirements is the  $\mathcal{H}_\infty$  robust control context with additional eigenvalue placement constraints. There exists an extensive literature over this general problem of *root clustering* (e.g. see [28, 56, 57]); here however the approach found in [27] will be preferred since the author believes that it gives the more general results on the subject. Having given the motivation why eigenvalue placement is so important in Section 3.1.1, this section presents some introductory material over the famous LMI regions.

#### 3.1.2.1 Design Objectives

As pointed out in the previous analysis, an eigenvalue placement procedure could be very efficient for a control system. This procedure could be either a rather exact or one-to-one eigenvalue assignment to predefined locations, or a more general placement of the system's state space representation eigenvalues into convex sub-regions of the complex plane  $P_{\mathbb{C}}$ . The latter method is very appealing because it can be cast as an LMI convex optimization problem solvable by efficient algorithms.

$\mathcal{D}(\alpha, r, \vartheta)$   
region

These regions may be vertical or horizontal strips, circles, parabolas or general conic sections on the complex plane. An LMI region used often in practice is the  $\mathcal{D}(\alpha, r, \vartheta)$  performance-stability region of Fig. 3.4. This particular LMI region could define a useful design objective as it is the intersection of an  $\alpha$ -stability vertical strip  $\mathcal{D}_\alpha$  that provides a minimum decay rate  $\alpha$ , a semi-circular region  $\mathcal{D}_r$  imposing undamped natural frequency constraints and a triangular constraint region  $\mathcal{D}_\vartheta$  that sets minimum damping on the closed loop eigenvalues. For any complex number  $z = x + yj \in \mathbb{C}$  these regions are defined as:

$$\mathcal{D}_\alpha : \quad \text{Re}\{z\} = x \leq -\alpha, \quad \alpha > 0 \quad (3.11)$$

$$\mathcal{D}_r : \quad |z| \leq r, \quad r > 0 \quad (3.12)$$

$$\mathcal{D}_\vartheta : \quad \tan \vartheta \cdot x \leq -|y|, \quad 0 < \vartheta < \pi/2 \quad (3.13)$$

and

$$\mathcal{D}(\alpha, r, \vartheta) \triangleq \mathcal{D}_\alpha \cap \mathcal{D}_r \cap \mathcal{D}_\vartheta. \quad (3.14)$$

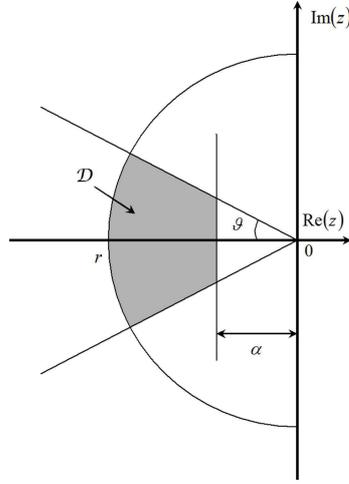


Figure 3.4:  $\mathcal{D}$  performance-stability region.

### 3.1.2.2 D-Stability

In order to use the powerful machinery of LMI solvers to confine the eigenvalues of a plant inside a given region  $\mathcal{D}$  of the complex plane  $P_{\mathbb{C}}$ , a formal definition of such a region is needed and it is given by the following statement [27]:

**Definition 3.1.** A subset  $\mathcal{D}$  of the complex plane  $P_{\mathbb{C}}$  is called an *LMI region* if there exists a symmetric matrix  $\mathbf{\Lambda}$  with  $\mathbf{\Lambda} = \mathbf{\Lambda}^T \in \mathbb{R}^{m \times m}$  and a matrix  $\mathbf{M} \in \mathbb{R}^{m \times m}$  so that:

$$\mathcal{D} =: \{f_{\mathcal{D}}(z) < 0, z \in \mathbb{C}\} \quad (3.15)$$

with:

$$f_{\mathcal{D}}(z) = \mathbf{\Lambda} + z\mathbf{M} + \bar{z}\mathbf{M}^T. \quad (3.16)$$

□

Given the negative definitiveness of Eq. 3.15 the LMI regions are always convex and symmetric with respect to the negative real axis of  $P_{\mathbb{C}}$  since  $f_{\mathcal{D}}(\bar{z}) = \bar{f}_{\mathcal{D}}(z)$ . In addition, more complex LMI regions may be constructed by simpler ones since they are in general invariant under set intersection<sup>3</sup>. This result was used for example in the previous section in order to construct the  $\mathcal{D}(\alpha, r, \vartheta)$  performance-stability region of Fig. 3.4 and will be further exploited when it comes to the placement of the eigenvalues of a LTI system inside this region.

Consider the following LTI and finite dimensional unforced system with  $x \in \mathbb{R}^{n \times 1}$  and  $\mathbf{A} \in \mathbb{R}^{n \times n}$ :

$$\dot{x} = \mathbf{A}x \quad (3.17)$$

<sup>3</sup>This means that the intersection  $f_{\mathcal{D}_1} \cap f_{\mathcal{D}_2}$  of two LMI regions is also an LMI region with  $f_{\mathcal{D}_1 \cap \mathcal{D}_2} = \text{Diag}(f_{\mathcal{D}_1}, f_{\mathcal{D}_2})$ .

The necessary and sufficient condition for the plant to be *quadratically asymptotically stable* is the following well-known Lyapunov inequality condition:

$$\exists \mathbf{X} = \mathbf{X}^T > 0 : \quad \mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{A}^T < 0. \quad (3.18)$$

The aforementioned condition may be extended for general *stable* subregions  $\mathcal{D}$  of the complex plane (LMI regions) as in Definition 3.1; if the spectrum of  $\mathbf{A}$  belongs to  $\mathcal{D}$ , then the system in Eq. 3.17 is called  $\mathcal{D}$ -stable. The following theorem gives necessary and sufficient conditions for  $\mathcal{D}$ -stability of such a system:

Stability  
condition  
for  
eigenvalue  
placement

**Theorem 3.1.** Consider the system of Eq. 3.17 and a convex LMI region  $\mathcal{D}$ , characterized by the matrices  $\mathbf{\Lambda}, \mathbf{M}$  and described by the complex function  $f_{\mathcal{D}}(z)$  as in Definition 3.1. Consider also the  $m \times m$  block matrix  $\mathbf{F}_{\mathcal{D}}(\mathbf{A}, \mathbf{X})$  with:

$$\begin{aligned} \mathbf{F}_{\mathcal{D}}(\mathbf{A}, \mathbf{X}) &= \mathbf{\Lambda} \otimes \mathbf{X} + \mathbf{M} \otimes (\mathbf{A}\mathbf{X}) + \mathbf{M}^T \otimes (\mathbf{A}\mathbf{X})^T \\ &= \left[ \Lambda_{kl}\mathbf{X} + M_{kl}\mathbf{A}\mathbf{X} + M_{lk}(\mathbf{A}\mathbf{X})^T \right]_{1 \leq k, l \leq m} \end{aligned} \quad (3.19)$$

The system in Eq. 3.17 is then called  $\mathcal{D}$ -stable if and only if there exists a matrix  $\mathbf{X} = \mathbf{X}^T > 0$  so that the following LMI condition holds:

$$\mathbf{F}_{\mathcal{D}}(\mathbf{A}, \mathbf{X}) < 0. \quad (3.20)$$

□

*Proof.* See [27], Appendix. ■

From the preceding analysis it is obvious that one could concatenate more than one LMI's of the form  $\mathbf{F}_{\mathcal{D}_i}(\mathbf{A}, \mathbf{X}) < 0$  for each  $i$ 'th LMI region; their intersection then forms the desired eigenvalue placement region of Eq. 3.14. This is exactly the power of the method since complex, performance-tailored LMI regions may be easily described in this way.

$\mathcal{D}(\alpha, r, \vartheta)$   
stability  
conditions

The corresponding LMI conditions for each of the  $\mathcal{D}(\alpha, r, \vartheta)$  subregions are given by the following expressions:

$$\mathcal{D}_{\alpha} : \quad \mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{A}^T + 2\alpha\mathbf{X} < 0 \quad (3.21)$$

$$\mathcal{D}_r : \quad \begin{bmatrix} -r\mathbf{X} & \mathbf{A}\mathbf{X} \\ \mathbf{X}\mathbf{A}^T & -r\mathbf{X} \end{bmatrix} < 0. \quad (3.22)$$

$$\mathcal{D}_{\vartheta} : \quad \begin{bmatrix} \sin \vartheta (\mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{A}^T) & \cos \vartheta (\mathbf{A}\mathbf{X} - \mathbf{X}\mathbf{A}^T) \\ \cos \vartheta (\mathbf{X}\mathbf{A}^T - \mathbf{A}\mathbf{X}) & \sin \vartheta (\mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{A}^T) \end{bmatrix} < 0. \quad (3.23)$$

This concludes the analysis concerning the conditions for eigenvalue placement inside LMI regions. In the following section the synthesis equations for the calculation of an output feedback  $\mathcal{H}_{\infty}$  controller with additional eigenvalue placement constraints<sup>4</sup> for the closed loop eigenvalues will be given.

<sup>4</sup>The regional constraints will be of the form as in Eqs. 3.21-3.23.