UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

TESI DI LAUREA MAGISTRALE IN
INGEGNERIA DELL'AUTOMAZIONE

FILTERING AND CONTROL ALGORITHMS FOR CRYSTAL PULLING PROCESSES

Relatore: Ch.mo Prof. Alessandro Beghi
Correlatore: Ch.mo Prof. Alessandro Chiuso

Laureando: Christian Spitaler

Padova, 23/04/2012

ANNO ACCADEMICO 2011 - 2012

Abstract

This master thesis addresses model-based state estimation and radius regulation for semiconductor crystals grown with the Czochralski process. These crystals are cut into thin slices (wafers), used for the fabrication of electronic micro devices. In industry currently radius control is achieved with PID controllers, not very appropriate for the time-variant, nonlinear Czochralski process. Thus more advanced regulation systems are required; for this purpose a nonlinear state space model, describing the hydro mechanical part of the process, was derived. Based on this model a linear-quadratic regulator (LQR) was designed. Furthermore for the estimation of the hidden state variables two statistical algorithms were compared, namely the Extended Kalman Filter and the Unscented Kalman Filter. These algorithms were found to yield equal statistical performance; furthermore the LQR approach works fine in simulation. So far no experimental closed loop tests were made.

Contents

Introduction

This master thesis addresses the problem of model-based state estimation and crystal radius regulation for semiconductor crystal growth processes, more precisely for the Czochralski process. It was developed during a practicum experience with MEMC Electronic Materials, an United States manufacturer of silicon wafers for the semiconductor industry based in St. Peters, Missouri. The practicum, supervised by Prof. Alessandro Beghi from the University of Padua, was performed from August 2011 to March 2012 at the MEMC plant situated at Merano, Italy.

The *Czochralski process* is a method to grow high-purity, possibly doped semiconductor crystals.[1] The device used for crystal growth is the *crystal puller*; an example is reported in Figure 1. Once the growth process is completed, the semiconductor crystals are cut into thin slices, the so-called *wafers*, which are used in electronics for the fabrication of integrated circuits and other microdevices [14]. As described in [14], the wafer is used as the substrate for microelectronic devices built in and over the wafer and undergoes many microfabrication process steps such as doping or ion implantation, etching, deposition of various materials, and photolithographic patterning. Furthermore solar cells and panels can be produced from semiconductor wafers.

A detailed description of the Czochralski process and its major difficulties is given in Chapter 1. Only the growth of *monocrystalline silicon crystals* for electronic wafers is considered, since these are the products fabricated at the Merano plant. Briefly high-purity silicon is melted in a quartz crucible, situated in the bottom part of the crystal puller. Normally a dopant (amongst others arsenic, antimony, phosphorus and boron) is added to the silicon melt. Then a monocrystalline silicon seed, attached to a rod, is dipped into the melt; the apex melts and a small, liquid region forms between the seed and the silicon melt contained in the crucible. This region is the so-called *meniscus*; its height is an important variable for radius control. Then the (rotating) seed is slowly pulled upwards, according to a pre-defined pull rate; in optimal conditions the silicon which solidifies at the interface between the seed and the meniscus is monocrystalline. How-

---

[1] the dopant is used to change the electronical properties of the semiconductor

ever at the moment of the contact between the seed and the melt a thermal shock occurs, causing undesired imperfections in the crystal structure, such as dislociations. In order to "grow them out" initially a small-diameter (for example 6 mm) portion is grown, the so-called *neck*. This part of the crystal is one of the most critical phases of the growth process. All structure imperfections have to be eliminated during neck, otherwise the structure of the whole crystal is compromised. Once the neck is completed, the crystal diameter is enlarged to the desired value according to a pre-defined mathematical function (this part of the crystal is the *shoulder*); then the constant radius part of the crystal, the *body*, is grown. The wafers are cut out only from the body.
Examples of a silicon seed, Czochralski grown crystals and wafers are reported in Figure 2.

Figure 1: Example of crystal puller [17] and schematic representation [18]

Figure 2: Monocrystalline silicon seed [15], crystals and wafers [16]

Czochralski crystal growth is an extremely complex, highly nonlinear, time variant batch[2] process [3]; a myriad of variables have to be taken into account in order to grow a monocrystalline crystal with the desired electronical (such as dopant and oxygen concentration) and structural properties, at the pre-defined diameter imposed by the wafer dimension. The low pressure (typically less than 100 mbar) atmosphere in the puller is enriched with argon; basically it is used to create an inert atmosphere to prevent oxidation on the crystal surface and puller components. Other critical parameters for crystal structure and impurity concentration are the strength of the magnetic field applied to the silicon melt and the crucible rotation speed [19] (usually both the crucible and the rod rotate).

According to [3] the most important parameters are the crystal growth rate (that is the solidification rate at the interface between crystal and meniscus) and the crystal radius. The growth rate affects the structural and electronical properties of the crystal, such as structure losses and dopant, oxygen and impurity concentrations in the crystal. To be exact the variable of interest is not just the growth rate, but the ratio of growth rate v_g to temperature gradient over the meniscus, G_m; this latter parameter depends on both the melt temperature and the meniscus height. Furthermore a precise radius regulation is required in order to reduce the material waste. Body portions with a too small diameter have to be remelted, whereas for portions with too large diameter material has to be cut off. Considering that the growth of a crystal continues for up to 2 days an imprecise radius regulation causes remarkable loss of time and material and thus it affects the production cost heavily.

Among these two variables only the radius can be measured, whereas the growth rate is a hidden variable. It is approximately equal to the pull rate and therefore the rates are often equated deliberately, in order to simplify the regulation; however this is not entirely correct. The radius is usually measured with a camera pointing to the meniscus; however to be exact the measurement refers to the bright ring which appears on the meniscus and therefore not to the true radius [3]. Thus the measurement is not very precise and quite noisy.[3] Usually the noise is filtered with a simple lowpass or tracking filter.

Typically the desired growth rate profile (which is chosen according to the electronical specifications for the crystal) is tracked acting on the power applied to the heater positioned around the quartz crucible.[4] One of the major problems of the *growth rate control loop* is the fact that it is characterized by a large time delay. In the first approximation the relation between the melt temperature, one of the variables which affect the growth rate, and

---

[2] that is, there exists no working point

[3] for a more detailed discussion refer to Chapter 1

[4] more precisely the crucible is contained in a graphite susceptor; the heater is positioned outside of this susceptor

the heater power is described by a first-order system with a time delay of approximately $20 \div 30$ minutes. Therefore this loop is very slow and the power variations have to be applied very carefully. The *radius (or diameter) control loop* is much faster; the radius is regulated with the pull rate. Variations of the pull rate are propagated almost instantaneously to the radius. A detailed description of the state of the art for radius and growth rate regulation is given in Chapter 2; typically simple PID controllers are used, although some model-based approaches are proposed in literature.

This thesis, more precisely the MEMC project it is a part of, has arisen from the necessity of more advanced, model-based control schemes for the ratio $\frac{v_g}{G_m}$ and the radius r, since PID control is clearly not the optimal choice for nonlinear batch systems as the Czochralski process [3]. The project is managed by Markus Siegert from the St. Peters plant. His work, heavily used and often cited in this thesis, is summarized in [1]; furthermore, details were discussed in private communications. From now on for these discussions the reference [2] is used.
To make the first move he developed a model-based control system for radius regulation. For this purpose a simple, nonlinear state space model, the so-called capillary model, was deduced; it relates the pull rate to the crystal radius. The three state variables are given by the radius, the meniscus height and the melt temperature; the output variable is the noisy radius measurement, whereas the pull rate is used as input variable. This model refers only to the geometrical part of the Czochralski process; the thermal part was not modelled, since it is extremely difficult to capture the thermal dynamics in the melt [3]. The thermal regulation, that is the growth rate control with the heater power, was not addressed so far; it is still achieved with standard PID control.
The meniscus height and the melt temperature are hidden, not measurable state variables, required by each kind of model-based state feedback controller. Therefore, based on the capillary model, a statistical state estimator, more precisely an Unscented Kalman Filter (UKF), was used in [1]; the state estimations computed by this filter are fed to an infinite-horizon linear-quadratic regulator (LQR) which computes the required pull rate according to the radius set point. The closed loop system was already tested on an actual puller, with satisfying results [1]. It is worth mentioning that so far only the neck phase was considered.
The collaboration with the University of Padua was started in order to solve some problems related with this approach. In the context of this thesis a detailed theoretical analysis of the work presented in [1] was performed: that is the modelling procedure, the LQR and the UKF design and tuning. The latter point is the principal reason for the collaboration with the University; statistical, nonlinear state estimation is a complex argument and for a detailed theoretical analysis and treatment highly specific skills are required.

In fact one of the major problems of the approach of Siegert is the filter tuning; as a matter of fact, he did not use any statistical methods for estimation evaluation and filter tuning. Since state estimation is fundamental not only for LQR but for each model-based control approach, the development of estimation quality indices for filter tuning is absolutely essential. Note that model-based control can not be performed without knowledge of the melt temperature and the meniscus height, thus the usage of an estimator for these hidden quantities is indispensable, since they can not be measured with sufficient precision.
Based on [1] this thesis provides:

- a detailed analysis of the deduction of the original capillary model;
- the development of an improved version of this model, with remarkable benefits in terms of robustness and linearity;
- analysis and simplification of the original LQR approach, reducing the number of state variables in the (improved) LQR model; thus the computational effort is reduced too;
- a detailed theoretical analysis of the nonlinear state estimation problem, based on the Bayesian approach. Besides the UKF, other estimation algorithms are exposed: the Extended Kalman Filter (EKF) and the Particle Filter (PF). These algorithms were implemented with the improved capillary model and compared to the original UKF approach;
- the development of numerical indices for state estimation quality evaluation, based on whiteness tests on the output prediction error;
- the development of filter tuning procedures based on these indices.

Like in [1] also the considerations of this report are limited to the neck phase; however they should be extendable, with some small adjustments, to the body phase too. The control and filtering algorithms developed during the practicum were implemented with *Matlab*, a numerical computing environment designed for complex matrix manipulations and statistical analysis, described more in detail in Chapter 7. Initially the algorithms were tested on the simulated capillary model and subsequently the simulation results were validated on experimental data provided by Siegert. This data was recorded during test runs on a puller at the St. Peters plant, performed with the original UKF and LQR approach. Obviously only the filtering algorithms were applied to this data, since state feedback control such as LQR can be tested only on the actual puller. During the practicum for lack of time it was not possible to execute such closed loop tests. The next step would be to implement the algorithms on an actual puller and verify their impact on the real system.

Briefly, the following results were obtained:

- the improved model is more linear and more robust with respect to model parameter variations, with benefits for both the state estimation and the LQR;
- the reduced state space LQR is simpler than the original approach and thus computationally more efficient. Furthermore some formal problems related with the original LQR were solved;
- an alternative filter approach, the EKF, was deduced. This state estimator is easier to handle than the UKF and yields approximately the same performance. Therefore it is a valid alternative to the UKF;
- it was proved that the probability density function of the state vector is almost Gaussian (this result was achieved with the Particle Filter). Thus the usage of more complex estimators (or other sigma point sets for the UKF) is quite senseless as long as the model is not complicated;
- a statistical index for state estimation evaluation was developed, which permits a precise tuning of the UKF (and EKF).

The report is structured as follows. Chapter 1 contains a general description of the Czochralski process and a typical crystal puller. Furthermore some basic definitions are given. Then in Chapter 2 a literature review on actual radius and growth rate regulation approaches is given (with some notes on state estimation).
Chapter 3 exposes the capillary model reported in [1], with a detailed description and deduction of the equations. Furthermore the model parameters are discussed. Finally some modifications and improvements are proposed, leading to the improved capillary model used heavily in the following. At the end of the chapter a brief stability and controllability analysis on the linearized improved model is performed.
Chapter 4 addresses the nonlinear estimation problem. After a general introduction of the Bayesian estimation approach adopted for this project, its recursive application to linear state estimation, the linear Kalman Filter, is exposed. Then its extensions to nonlinear estimation, the EKF and the UKF, are described in detail, together with a more advanced approach, the Particle Filter. This latter filter was used only for some verifications on the probability density function of the state space vector, useful for UKF and EKF estimation.
Chapter 5 introduces a widely used statistical estimation quality test, the so-called test for white noise. The illustration of the underlying principle is followed by the definition of some basic functions (autocorrelation, Power

Spectral Density and Power Spectrum, besides white noise) required for this test, together with a parametric and a non-parametric approach to their estimation. Finally two simple numerical quality indices are deduced.
In Chapter 6 some theoretical considerations on the LQR approach are reported. The application to radius regulation as in [1] is discussed and after that some improvements are proposed, leading to the reduced state space LQR with feed forward. Finally some considerations on the penalty choices are given.
Chapter 7 discusses the simulation results. Initially a brief description of the implementation of the capillary model, the filtering and the control algorithms in Matlab is given. Then the performance of the original UKF used in [1] is compared to the UKF and EKF based on the improved capillary model. The filter tunings and comparisons are based on the white noise area index. Finally the analysis is completed with the Particle Filter. The chapter is concluded with a brief robustness discussion and some considerations on the improved LQR approach.
In Chapter 8 the state estimation algorithms (old and improved UKF and EKF) are applied to experimental data, in order to verify the simulation results.
Finally in the last chapter the results are summarized and some conclusions are drawn. Furthermore, some suggestions for future work are reported.

In the following some critical informations and results are omitted, since their disclosure is prohibited by the confidentiality agreement with MEMC Electronic Materials.

Chapter 1

The Czochralski process

The Czochralski process is a method to grow high-purity, monocrystalline semiconductor (and metal) crystals, which are then cut into thin wafers, used in electronics for the fabrication of integrated circuits and furthermore for the production of solar cells and panels. Hereinafter the focus is on monocrystalline silicon crystals for electronic wafers. The electrical properties of the wafers depend on the crystallography, the impurity and the dopant concentration in the crystal.
In this chapter initially a brief historical overview on Czochralski growth is given. Then two big issues of the growth process are described and discussed: crystallographic defects and impurities. Finally the structure of a modern puller is considered and the Czochralski pulling process is explained briefly.

Figure 1.1: Czochralski crystal puller plant [39]

1.1 Historical overview

The following overview is based on [20].
The Czochralski process is named after Polish metallurgist Jan Czochralski. In 1916 he accidentally discovered that dipping a monocrystalline tin seed into molten tin it is possible to grow monocrystalline tin wires. Czochralski used his discovery to measure the crystallization rate of metals. His method was first used to grow semiconductor crystals, namely germanium crystals, by Teal and Little from Bell Labs, in 1948; the first silicon crystal was grown in 1950. They also equipped their puller with devices for doping. Initially the biggest problem was the low purity of the starting polysilicon melted in the crucible. Since the 1960s the purification and refining of polysilicon is obtained transforming the raw silicon into a silane, more precisely trichlorosilane, that is a colorless, toxic and extremely flammable gas with chemical formula HCl_3Si. The silane is purified by distillation and adsorption and finally retransformed into elemental silicon by chemical vapor deposition. The polysilicon obtained this way is sufficiently pure for Czochralski crystal pulling; that means that the total concentrations of impurity elements (other than the electrically inactive oxygen and carbon, which are allowed to occur in higher concentrations) are below 1 ppba.[1]
Another important step was the development of techniques for dislocation-free crystal growth, first achieved in 1958. Dislocations are line defects in the crystal structure. Furthermore over the years also the swirl-like microdefects and the number and size of voids in the crystal were reduced. A brief introduction to these crystallographic defects is given below. For electronic wafers perfectly monocrystalline, dislocation-free crystals are required; beside purity constraints this is the most important demand. In the 1970s also the important role of oxygen in the silicon crystal was recognized; basically it can be used to impede the movemement and multiplication of dislocations and to bond impurities.
The first crystals had a diameter of about $1 \div 2$ cm and weighed less than 100 g. Today crystals up to a length of 2 m, a weight of hundreds of kilograms and a diameter of 450 mm are grown. More than 95% of the single[2] silicon crystals are grown with the Czochralski technique, whereas the rest is grown by the float zone approach (that is a crucible-free method for crystal growth, not considered in this work).

1.2 Crystallographic defects

In the following the most important crystallographic defects encountered in Czochralski silicon crystal growth are described briefly. Crystalline solids

---

[1] parts per billion atomic

[2] that is, monocrystalline

such as silicon exhibit a regular crystal structure with a pre-defined orientation. However the structure or lattice is not perfect since it is affected by different types of defects, namely:

- Point defects [21]
  Point defects are places where an atom is missing or irregularly placed in the crystal lattice; they are always present. If an atom is missing in the lattice the defect is called vacancy; irregularly placed atoms are called interstitials. Basically two different types of interstitials can appear: if the misplaced atom is a silicon atom it is referred to as a self-interstitial defect; if it is of different type it is called an impurity interstitial. Impurity interstitials are often introduced deliberately in order to modify the properties of the crystal. Furthermore impurity atoms can occupy a regular place in the lattice, forming a substitutional defect (the impurity atom replaces a silicon atom).
  Figure 1.2 shows a vacancy in the silicon lattice (the vacancy is represented with an empty circle). Both the vacancies and the interstitials can move through the crystal; the speed of motion depends on the temperature gradient. Point defects can form agglomerates; vacancy agglomerates are often referred to as voids. Furthermore they can disappear if an interstitial encounters and occupies a vacancy.
  The motion of the vacancies and the interstitials is governed by the ratio $\frac{v_g}{G_m}$. Vacancies can be used to eliminate line defects [22].

- Line defects or dislocations
  According to [21], linear defects are groups of atoms in irregular positions; usually they are called dislocations. Dislocations cause lattice strain and distortion; basically plastic (that is permanent) deformation is caused by the motion of dislocations. Two different types can be distinguished: edge and screw dislocations. Furthermore often hybrid dislocations occur. An example of an edge dislocation is reported in Figure 1.2; basically it is an extra half-plane of atoms in the lattice. Screw dislocations are more difficult to represent; intuitively they can be described as helical irregularities in the lattice. Dislocations can move (or slip) because of cooling strains in the crystal; they are generated due to thermal stress, which in the case of Czochralski crystal growth is most pronounced at the moment of the contact between the single silicon seed and the melt.
  According to [20], for most lattice orientations the dislocations are lying oblique to the crystal growth direction and therefore it is possible to “grow them out“ to the lateral crystal surface; thus they are left behind by the growth interface and eliminated from the crystal. However they can slip and therefore follow the crystallization interface. To prevent this undesired phenomenon usually Dash seeding is used: that is, after the seed was dipped into the melt the growth

is started reducing the diameter, which is then kept constant for a certain length (this portion of the crystal is the neck), at an increased growth rate. For example, the seed could have a diameter of 10 mm which is then reduced to 5 mm. Due to the small diameter the cooling strains are reduced and therefore the dislocations are left behind. Once this was achieved the diameter is enlarged (shoulder phase) to the desired value for the wafers. In a dislocation-free crystal new dislocations are introduced only if very high cooling strains appear; however, if this happens, they are multiplied extremely fast. If not all dislocations are eliminated during neck or new dislocations form after this phase the structure of the whole crystal is compromised.
As stated above dislocation multiplication and movement can be reduced by adjusting the oxygen concentration in the crystal. Furthermore the larger the diameter of the crystal, the higher are the cooling strains and therefore the risk of formation of new dislocations increases. One possibility to overcome this problem is to stabilize the thermal convection flows in the melt with magnetic fields [20].

- Swirl-like microdefects [20]
  Swirl-like microdefects are concentric agglomerates of point defects which occur only in dislocation-free crystals. They are caused by growth rate fluctuations.

Furthermore plane and bulk defects can occur, however the discussion of these defects is beyond the scope of this paper (refer to [21] for a detailed description). Crystallographic defects affect the physical, chemical and electronical properties of the crystal and therefore it is very important to take them under control.

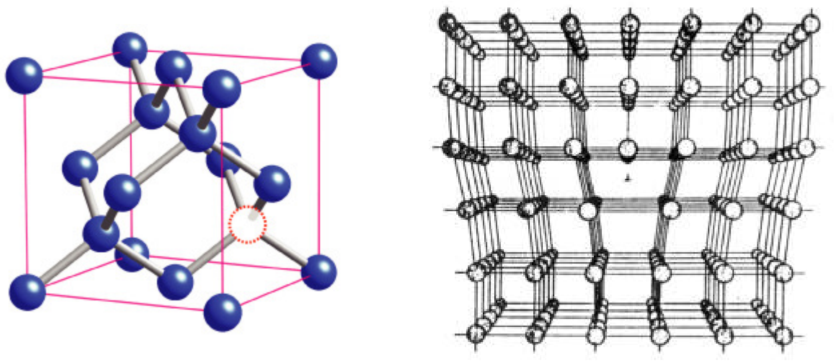

Figure 1.2: Vacancy in silicon lattice [40] and edge dislocation [41]

1.3 Impurities

As stated at the beginning of this chapter, one of the major concerns in Czochralski crystal growth, besides the reduction and elimination of crystallographic defects, is the purity of the silicon. According to [20] the impurity concentration has to be lower than 1 ppba. Only carbon and oxygen are allowed to appear in higher concentrations. It is necessary to distinguish between undesired impurities and intentionally introduced impurities (such as dopants). Only the former ones are impurities in the proper sense and therefore considered in this section.

With the introduction of new purification techniques for polysilicon (based on trichlorosilane distillation and adsorption) the impurity problem was partially solved. However impurities can be introduced also during the pulling process, mainly from the crucible into the silicon melt. The crucible, supported by a graphite susceptor, is usually made of quartz glass, more precisely silica, with the chemical formula SiO_2 (silicon dioxide). At high temperatures it becomes soft and deforms, and furthermore it reacts with the silicon melt [20]; thus the impurities contained in the crucible are transmitted to the melt. For this reason it is important to use very pure materials for the crucible; thus syntethic high-purity silica is utilized more and more.

Besides the impurities the reaction with the silica crucible introduces also oxygen into the melt [20]. Fortunately oxygen evaporates easily in the form of silicon monoxide (SiO), thus preventing oxygen supersaturation of the melt, which leads to dislocated and finally polycrystalline growth.[3] With any other crucible material[4] this strong evaporation of the introduced impurities is not given, since they form no volatile compounds with silicon, thus remaining in the melt and leading to supersaturation [20].

With silica crucibles more than 99% of the oxygen introduced in the melt evaporates at the surface.[5] However, if it is not removed from the puller atmosphere, a slag of SiO, SiO_2 and Si forms at the melt surface, thus compromising the growth of the single silicon crystal. Therefore the atmosphere has to be purged; this is usually achieved with some noble gas, typically argon [20].

Figure 1.3 reports an unused and an used silica crucible. Note that actually two unused crucibles are shown: the smaller one is placed in the bigger one (obviously only one crucible is placed in the puller). The used crucible contains a residual of solidified silicon, not consumed during crystal growth.

---

[3] this happens for each kind of supersaturation, not just with oxygen

[4] such as silicon nitride or silicon carbide

[5] and, as stated previously, a small amount of oxygen in the silicon can be very useful to treat dislocations and impurities

Figure 1.3: Unused and used silica crucibles [42]

1.4 Crystal puller and pulling process description

Figure 1.4 reports the schematic of the lower chamber of a modern crystal puller. The quartz crucible is charged with crushed high-purity polysilicon and placed in the puller. The silicon is then melted with a graphite heater. Since the quartz crucible deforms and becomes soft at the high temperatures required for silicon melting, it is supported by a graphite susceptor (or graphite crucible), placed on a rotatable shaft. The entire hot zone (that is the lower part of the puller, used for heating) is insulated with thermal shields. Furthermore, typically a radiation shield, not shown in the figure, is used to shield the crystal from the melt. Its position is adjustable and represents a critical parameter for the temperature gradients in the hot zone. Without such a reflector it is very difficult to grow large diameter crystals [20].

As mentioned previously inside the puller a low-pressure inert gas atmosphere is generated; normally a noble gas such as argon is used. For this purpose the puller is equipped with an elaborated system of vacuum pumps and valves.

The single silicon seed is mounted on the so-called seed holder connected to a rotatable rod or cable; a spindle, positioned at the top of the puller, is used to pull the crystal up. The upper chamber with the winch, not illustrated in the figure, can be segregated from the lower chamber. As a matter of fact, once the growth process is completed, the crystal is pulled up, the valve between the chambers is closed and after the cooling procedures the front door of the upper chamber is opened and the crystal can be removed by the operator. Theoretically once the puller was charged, the whole pulling process runs fully automatic until cooling of the grown crystal is completed; therefore during growth the operator can easily supervise several pullers.

Furthermore modern pullers are equipped with strong electromagnets, used

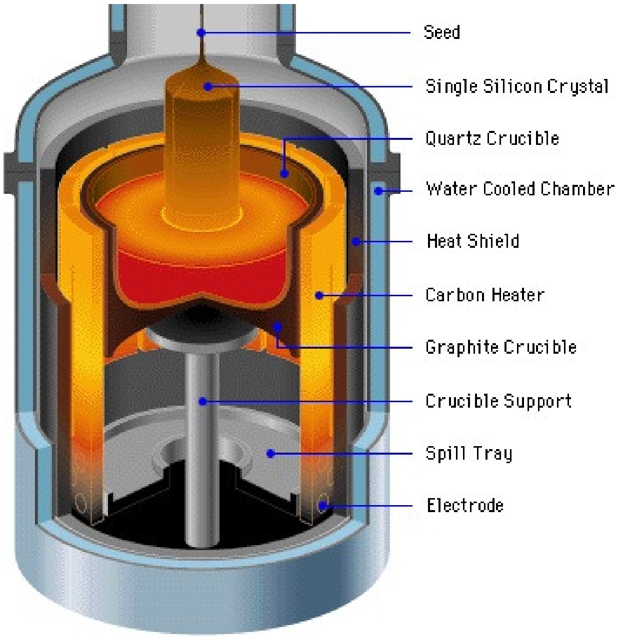

Figure 1.4: Czochralski crystal puller schematic [43]

to create magnetic fields inside the crucible. The magnetic field can slow the thermal convection flow in the melt; however it is primarily used to stabilize it, that is to damp the fluctuations of the flow [2]. Different magnet field configurations are used for this purpose; furthermore often a secondary magnet is positioned besides the radiation shield. The magnets are not shown in the schematic.
Moreover the lower chamber is assembled with a viewing port for the operator and an optical system for crystal radius measurement, typically one or more cameras. The principle of the radius measurement is briefly explained below. Furthermore usually at the top of the upper chamber a pyrometer, pointing to the silicon melt, is mounted.

Finally a brief summary of the different phases of the Czochralski pulling process is given. The main steps are schematized in Figure 1.6.
First the crucible is charged with crushed polysilicon and eventually some dopant.[6] Then the crucible is placed in the graphite susceptor and the lower chamber is closed. After that the puller is sealed and the air is pumped out with a vacuum pump, in order to generate a vacuum (the atmosphere is in the following enriched with argon). Then the heater is turned on and

---

[6] such as boron, arsenic, antimony or phosphorus. Dopants are used to modify the electrical properties, more precisely the final resistivity, of the crystal; furthermore modern pullers are equipped with devices for redoping during the process, the so-called feeders

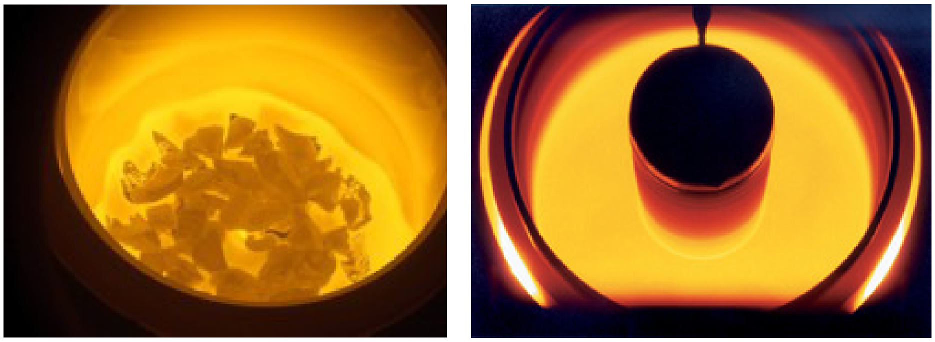

Figure 1.5: Viewing port perspective during melting [44] and body growth [45]

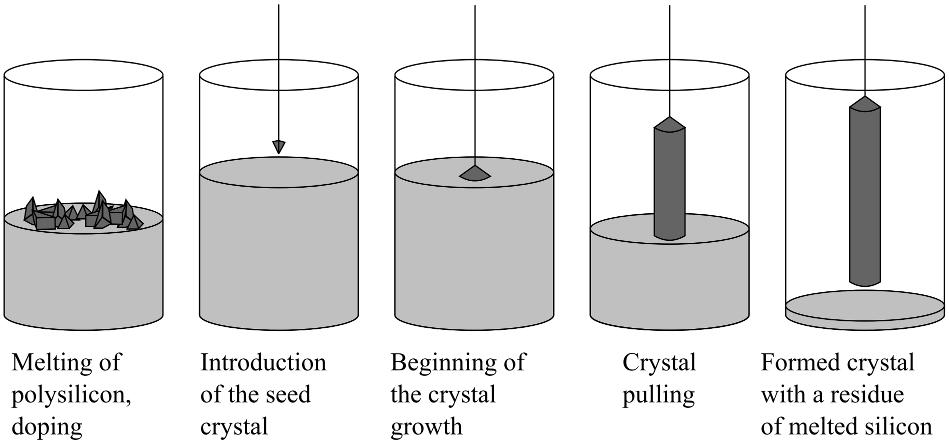

Figure 1.6: Schematic representation of Czochralski process [46]

the silicon is melted at a temperature of approximately 1500 °C. During the melting phase, as well as during the following phases, the crucible is rotated; for melting the rotations are used in order to mix liquid and solid silicon thoroughly.

Once melting is completed the temperature is stabilized at the desired value for crystal growth; this is achieved by adjusting the heater power. Temperature stabilization is followed by the dipping of the single silicon seed into the melt. In order to eliminate the dislocations, generated by the thermal shock at the moment of the contact, a neck is grown (according to the technique of Dash, described previously). During neck as well as during the following phases both the crucible and the crystal are rotated, usually in opposite directions. According to [19] crucible rotation enhances the thermal convection flows in the silicon melt, which are generated by non-vertical temperature gradients. Basically the crucible rotations are used to

homogenize the temperature distribution in the melt [2]. Furthermore the crucible can be lifted in order to compensate for the melt level reduction, which occurs due to the progressive solidification of the liquid silicon.
Once the neck is completed, the crystal diameter is enlarged to the desired value according to a pre-defined mathematical function (shoulder); then the constant radius body of the crystal is grown. As the body is completed the pull rate is increased; thus the radius starts to decrease and the end cone of the crystal is formed. The end cone is grown in order to avoid thermal shocks at the moment of the extraction. Finally the crystal is placed in the upper chamber and the cooling procedures are started; then the cooled crystal is removed.

The chapter is concluded with the discussion of a very important point of the pulling process and the definition of some useful terms and variables.
As mentioned in the introduction, when the seed is withdrawn from the melt, a small amount of molten silicon rises with it, pulled upward by surface tension; this phenomenon is well known as capillarity. The shape and especially the height of this small region of liquid silicon between the crystallization interface and the melt, usually referred to as *meniscus*, are of crucial importance for radius control ([2], [3]). This will become clear in Chapter 3, where the capillary model derived in [1] is described. The term "capillary" is used because the model describes the geometrical aspects of the meniscus, generated by capillary actions, as pointed out above.
The interface between the (liquid) meniscus and the solid crystal is the so-called crystallization or *growth interface*. Its shape is controlled by the temperature gradients in the hot zone; however usually for modelling purposes it is assumed to be perfectly flat (a condition which in practice is never satisfied). The rim of the interface is the so-called *trijunction line*: it is the intersecting line between solid and liquid silicon and the puller atmosphere. The *crystal growth rate* (or crystallization or solidification rate) is usually considered at this line and is defined as:

$$v_g = \frac{dl}{dt}.$$

l is the length of the crystal; thus the growth rate is the time derivative of the crystal length. Its value depends on the considered point of the trijunction line; however usually this dependence is neglected in Czochralski process modelling. The *pull rate* v is the speed at which the crystal is pulled out from the melt and its dynamic *does not equal the dynamic of the growth rate*; this point will be considered more in detail in Chapter 3. However the *average* pull rate must equal the *average* growth rate, since otherwise the meniscus height would diverge, thus impeding the growth of the crystal at the desired, constant radius [2].
The growth rate is one of the most important parameters in Czochralski

crystal growth and affects almost all the chemical and structural properties of the crystal. Thus, once the optimal profile was chosen, it has to be tracked carefully. The growth rate depends on the temperature gradients in the meniscus and in the crystal and therefore it is governed by the thermal properties of the process; typically it is regulated by adjusting the heater power.
Finally the actual *crystal radius* is defined as the radius of the crystallization interface (it was tacitly assumed to be of perfectly cylindrical shape, another simplification which is not necessary verified). However the camera measurement of the crystal radius is based on the bright ring resulting from the reflections of hotter locations in the hot zone by the meniscus [3], and therefore it does not necessary refer to the true radius. Amongst other problems it is difficult to establish whether the measurement refers to the top of the meniscus, as it should, or not.

Summarizing Czochralski crystal growth is an extremely complex, highly nonlinear, time variant batch process. The structural, chemical and electronical properties of the crystal are affected by hundreds of variables. Here only some selected properties and parameters were considered, since the goal of the project is model-based estimation and regulation for radius tracking and for this purpose a detailed analysis of the chemical and physical properties is not required. For a detailed, complete treatment of the properties and difficulties of Czochralski crystal growth refer, for instance, to [19].

Chapter 2

State of the art: radius and growth rate control approaches

This chapter reports a brief literature review on crystal radius and growth rate regulation approaches for the Czochralski growth process. Although this project addresses only radius control, also growth rate regulation was reviewed, since the two variables are mostly considered together. Furthermore some considerations on state variables estimation, required for model-based control, are listed.
At present, the standard for radius and growth rate regulation in industrial pullers consists in a double (or sometimes cascaded) PID feedback loop [49], as reported in Figure 2.1. r denotes the radius, v_g the growth rate, v the pull rate and P the heater power. This control method is typically adopted without any kind of model of the process dynamics; therefore the PID tuning is highly empirical and basically obtained by a trial-and-error procedure.

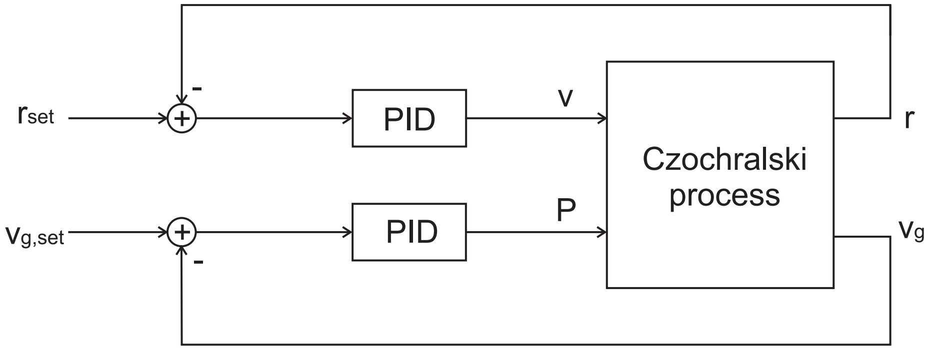

Figure 2.1: Standard industrial PID control of crystal radius r and growth rate v_g

The radius is regulated acting on the pull rate, whereas the growth rate is controlled through the heater power. Note that the second control loop is characterized by a large time delay, as discussed previously; more generally the dynamics of the growth rate control loop, that is the thermal dynamics of the process, are much slower than the dynamics of the radius control loop and therefore it operates usually on a larger time scale. The figure is just a schematic representation; obviously in a real control scheme the measured variables, usually the radius, have to be filtered, whereas the hidden variables, such as the growth rate, are estimated by some estimator or observer. Often the scheme is further simplified, that is the second loop refers to the pull rate and not to the growth rate. With this approach the loops work in cascade and no estimator for the growth rate is required. However this is not entirely correct since $v \neq v_g$. The scheme is often combined with feed forward terms for both the pull rate and the heater power (deduced from previous runs or according to some other criterion).
However PID regulators are linear controllers and therefore clearly not the optimal choice for a nonlinear, time variant batch process such as Czochralski crystal growth [3]. For such processes the tuning of the PID controllers is very tricky and basically obtained by trial-and-error, especially if no model of the system dynamics is available; thus the performance of such an approach is limited. Model-based control is preferable in this context; however modelling of the overall Czochralski process is very difficult, due to its complicated heat transfer mechanisms (such as the convection flows in the silicon melt), and requires a distributed parameter approach [3]. Gevelber et al. [47] made the first steps towards model-based control with conventional PID controllers, deriving a conduction-based lumped parameter model of the overall process. However no experimental results are available for his approach; anyway it can be stated, as pointed out in [3], that such a lumped parameter model of the *overall process* would not be robust enough for feedback control. Therefore most of the model-based control approaches proposed in literature consider a lumped parameter model only of the geometrical-hydromechanical properties of the meniscus, which can be used for the radius control loop;[1] usually the thermal dynamics of the process, which are extremely complicated, are not modelled, and growth rate control is achieved with conventional PID regulators. In the following some selected approaches are described.

An interesting approach was proposed by Mutti and Voronkov [48]. It is based on a cascaded radius and growth rate control scheme and uses the pull rate respectively the heater power as control variables, as usual. For this purpose they derived a simple model which describes the geometrical aspects of the meniscus, relating the measured crystal radius $y(t)$ to the

---

[1] such as the capillary model in [1]

true radius, the meniscus height and the growth rate.

According to [48] the radius measurement variations are recorded for a pre-defined interval of time, the so-called observation interval; during this interval the controller does not act on the system, that is the measurements are recorded in the open loop system. At the end of the interval the recorded radius function $y_{exp}(t)$ is compared to the function $y(t)$, parameterized with the true radius, meniscus height and growth rate; thus the current values of these parameters are deduced with a best fit routine applied to $y(t)$ and $y_{exp}(t)$. Based on these values a height response coefficient A_h and a power response coefficient A_P are determined; these parameters are then used for the computation of the required pull rate respectively heater power variations, in order to track the desired crystal radius and growth rate trajectories. This approach is an elegant compromise of model-based and empirical control (used for the power adjustments, since no thermal model was derived). However no experimental results are available.

A more advanced approach was proposed by Winkler et al. [3]. As well as Mutti and Voronkov they do not consider a complex mathematical model of the overall process, which would not be exact and robust enough for feedback control. Only the hydromechanical-geometrical part of the process is modelled, deriving a nonlinear state space model parameterized in crystal length instead of time; basically it describes the geometrical properties of the meniscus. The state variables are the crystal radius and the crystal slope angle (this variable will be defined in the following chapter); as control variable they define the so-called lift ratio, which takes into account the pull rate, the growth rate and the crucible translation rate. It is worth mentioning that the model was derived with the assumptions of cylindrical symmetry of the growth system and flatness of the growth interface.

Starting from this model Winkler designed a nonlinear model-based controller, based on flatness considerations;[2] this controller is used, in combination with PID controllers and feed forward actions, to regulate the radius. The thermal part of the process, that is the growth rate, is still regulated with conventional PID control acting on the heater power, since no thermal model was derived.

Moreover [3] addresses also the problem of hidden state variable estimation, namely of the crystal slope angle and the radius; furthermore also the growth rate is computed from the estimation of these variables. For this purpose the force acting on a load cell mounted at the top of the pulling rod is considered; from the first time derivative of this signal Winkler deduces, with a nonlinear model-based estimator, the values of the hidden

---

[2] as a matter of fact, their nonlinear model is flat; intuitively flatness is the extension of the controllability property of linear systems to nonlinear systems

variables. It is worthwhile to stress once more that also the radius is considered as a hidden, not directly measured variable.
The usefulness of the approach, theoretically applicable to all phases of the growth process (neck, shoulder and body), was proved with several experimental results obtained from the growth of gallium-arsenide and indium-phosphide crystals; however it was not yet applied to silicon crystal growth.

The approach of Winkler is very similar to the approach proposed in [1] and used for this project, that is LQR control of the radius, combined with UKF estimation of the hidden state variables. Both methods model only the geometrical part of the process and regulate the growth rate with a conventional PID loop. Furthermore the geometrical models are similar, besides the choice of different state variables. However, the model in [1] is parameterized in time and not in crystal length, and the measured variable is the radius and not the force acting on the load cell. Moreover LQR is a linear control approach whereas Winkler considers an ad hoc-designed, flatness based nonlinear controller, and finally his state estimation approach is deterministic and not statistical as the UKF method. Summarizing it can be stated that Winkler proposes a more complex approach, which can not be applied in absence of a load cell. On the other hand the strategy described in [1] requires only the radius measurement, certainly available on modern pullers for Czochralski crystal growth.

Finally an even more advanced approach was described by Irizarry-Rivera and Seider in [49] and [50]. They use nonlinear model-predictive controllers (MPC), operating on different time scales, for both the radius regulation and the growth rate regulation. The control variables are the pull rate respectively the heater power, as usual. However to be precise actually the second loop refers to the pull rate and not to the growth rate; thus the two rates are equated deliberately, as often done in crystal pulling. For the radius MPC loop (denoted as capillary loop) a geometrical state space model for the meniscus is derived, just as in [1] and [3]; however also the possibility of a non-flat crystallization interface is taken into account (by piecewise linearization). The state variables are the (true) crystal radius and the meniscus height, which are computed indirectly from the growth rate estimation; the measured signal is the crystal radius. Moreover the MPC law includes constraints on the pull rate and its variations.
Furthermore a thermal model for the growth rate loop (denoted as bulk loop) is considered. This model was obtained by discretization, using the boundary-element method for order reduction, of a distributed-parameter model which describes the heat transfer in the melt and the crystal. In [49] this thermal model is derived considering only the heat transfer by conduction; in [50] it was extended to the convection flows in the silicon melt.

Moreover also the thermal MPC law includes a set of constraints, basically on the control variables and on the heat flow variations (in order to reduce thermal stresses). The parameters of the thermal model are adjusted online, based on temperature measurements in certain critical points of the growth system, for instance in the crucible. However even with modern technology it is not possible to make temperature measurements in the silicon melt: in their simulations Irizarry-Rivera and Seider replaced the measurements with simulated values.

For this approach only simulated results are available. The principle of a coupled model-predictive control of both the radius and the growth rate is certainly interesting; however no experimental results are available, and furthermore the simulations refer to unrealistic conditions for industrial crystal pulling: Seider and Irizarry-Rivera consider a crucible with a diameter of 16 cm! Finally the reduced-order convection model for the growth rate control loop requires temperature measurements which are not available as matters stand. Therefore this approach, interesting from a theoretical point of view, has to be revisited.

Chapter 3

Deduction of the capillary model in state space form

3.1 Generalities

As pointed out previously, this project aims to the model-based regulation[1] of the crystal radius, acting on the pull rate. This part of the control scheme is called the *radius control loop*. Further on, as discussed in the previous chapters, the growth rate has to be regulated; this can be achieved by acting on the heater power of the puller. This second loop, the so-called *growth rate control loop*, is not investigated in this work. For the purposes of this project it is sufficient to use the simple, already implemented PID controller for the growth rate regulation.

In order to achieve an improved, model-based control of the radius, a model which relates pull rate v to crystal radius r is required. For this purpose, the so-called capillary model, described in [1], was used as a starting point (it was derived for neck conditions, but most probably it applies to body conditions too, with some small adjustments; however this was not yet investigated). In the following sections this model is presented and explained, with a detailed description of the modelling procedure. After that, some improvements are added, in order to obtain a simpler, more robust model. In order to distinguish it from the *original* model reported in [1] it is referred to as the *improved* model.

---

[1] and estimation of the required variables

3.2 The original capillary model

[1] proposes a simple, discrete time model in state space form, which relates the crystal radius to the pull rate. The state update equations, reported below in compact form, refer to Figure 3.1; the meaning of the variables and parameters is summarized in Table 3.1. In order to facilitate the readability, the time argument $n\Delta t$, where Δt is the sample time, was replaced by n. Anyway, in the following the sample time is always assumed to be 1 second.

$$\begin{cases} x_1(n+1) = f_1(x(n),\, u(n)) \\ x_2(n+1) = f_2(x(n),\, u(n)) \\ x_3(n+1) = f_3(x(n),\, u(n)) \end{cases}, \tag{3.1}$$

where:

$$x(n) = \begin{bmatrix} r(n) \\ h(n) \\ T_m(n) \end{bmatrix}, \qquad u(n) = v(n).$$

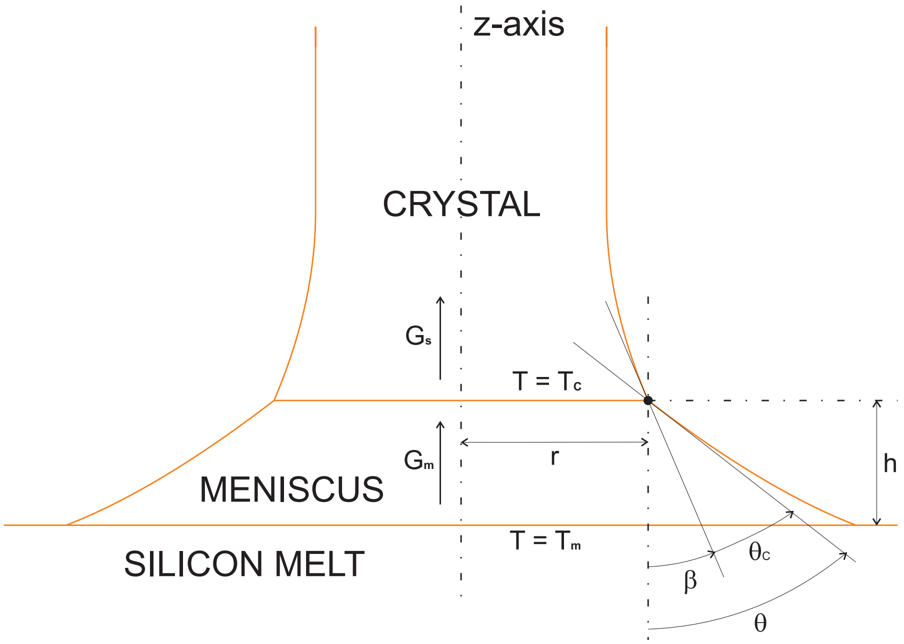

Figure 3.1: Reference scheme for the capillary model

Parameter / variable	***Symbol [unit]***
Crystal radius at the trijunction line*	$r\ [m]$
Meniscus height at the trijunction line*	$h\ [m]$
Melt temperature under the meniscus*	$T_m\ [K]$
Pull rate*	$v\ [\frac{m}{s}]$
Growth rate at the trijunction line*	$v_g\ [\frac{m}{s}]$
Meniscus slope angle	$\theta\ [rad]$
Sample time	$\Delta t\ [s]$
Melting point temperature of silicon	$T_c\ [K]$
Thermal conductivity of solid silicon	$\lambda_s\ [\frac{W}{mK}]$
Thermal conductivity of liquid silicon	$\lambda_L\ [\frac{W}{mK}]$
Density of solid silicon at T_c	$\rho_s\ [\frac{kg}{m^3}]$
Density of liquid silicon at T_c	$\rho_m\ [\frac{kg}{m^3}]$
Specific latent heat of fusion	$\Delta H_f\ [\frac{J}{kg}]$
Capillary or Laplace constant	$a\ [m]$
Equilibrium growth angle	$\theta_C\ [rad]$
Crystal slope angle*	$\beta = \theta - \theta_C\ [rad]$
Temperature gradient over the crystal	$G_s\ [\frac{K}{m}]$
Temperature gradient over the meniscus*	$G_m\ [\frac{K}{m}]$

Table 3.1: Model variables (denoted by *) and parameters

Briefly, the crystal radius r at time step $n+1$ is obtained from its previous value by adding a term which depends on the pull rate v and the tangent of the so-called crystal slope angle $\beta = \theta - \theta_c$; as can be seen from Figure 3.1, β is the angle between the vertical and the tangent to the crystal surface at the trijunction line. The equilibrium growth angle θ_C is a material property, however the meniscus slope angle θ (the angle between the vertical and the tangent to the meniscus surface at the trijunction line) is a function of r and

the meniscus height h.
Furthermore, the second state variable h depends on its previous value and an additional term determined by the difference between the pull rate and the crystal growth rate v_g; the pull rate is considered as the (only) input variable, whereas the growth rate *at the trijunction line* is a function of h and T_m:

$$v_g = v_g(h(n),\ T_m(n)) \tag{3.2}$$

As will be discussed later, almost all of the parameters which appear in this formula are difficult to determine and therefore affected by huge uncertainties.
The third state equation models the melt temperature as a random walk;[2] this is achieved resorting to an additive, Gaussian distributed white noise process with zero mean. If the power applied to the heater is constant, also the melt temperature is approximately constant and therefore no white noise term is necessary; anyway, as mentioned earlier, beside the radius control loop there is also the growth rate loop acting on the puller, which adjusts the heater power in order the achieve the desired growth rate. The consequent melt temperature evolution can be qualitatively obtained by simply adjusting the variance of the noise term.

Finally, the output equation of the state space model is:

$$y(n) = r(n) + w_y(n).$$

As a matter of fact, the only measured variable is the first state variable, i.e. the crystal radius.[3] Obviously this measurement is affected by measurement noise, represented by $w_y(\cdot)$. This noise term is assumed to have the same properties as the one acting on the third state variable.

3.2.1 Deduction of the model equations

This subsection contains a detailed analysis of the original capillary model, with a step by step deduction of the state update equations. The model is based on two fundamental assumptions, which are supposed to hold throughout the whole paper, namely:

i) Cylindrical symmetry of the whole system. More precisely, all properties of the system are supposed to be invariant with respect to rotations around the z-axis in Figure 3.1.

ii) Perfectly planar interface between the meniscus and the crystal, i.e. between the molten and the solid silicon. From a theoretical point of

---

[2] this strategy is adopted only for simulation

[3] measured by the camera

view this is equivalent to consider the radial temperature gradient on the crystallization interface equal to zero.

These two simplifications are quite standard in the context of capillary crystal growth [3]. With reference to Figure 3.1, lets consider the crystal radius update equation.

Crystal radius update equation

The first state update equation is based on the following fundamental relation between the radius derivative $\dot{r}$, the growth rate v_g at the trijunction line and the crystal slope angle β:

$$\frac{d\,r}{d\,t} = \dot{r} = v_g\ tan(\beta) = v_g\ tan(\theta - \theta_c).$$

This equation is of common use in capillary modelling (see [3] and [6]); in [7] a formal proof is given. Briefly, the proof is based on the assumption of thermodynamic equilibrium on the trijunction line; as a matter of fact, only under this condition the equilibrium growth angle θ_C can be considered constant. Note that the meniscus slope angle θ depends on r and h, as mentioned earlier.
In order to obtain the expression for the radius its derivative is approximated with the incremental ratio.

Growth rate v_g

Formula (3.2) for the crystal growth rate v_g on the trijunction line is based on the heat flow balance on the interface between the meniscus and the crystal; considering only the heat flows due to thermal conduction, neglecting the radiative heat transfer, yields [3]:

$$\underbrace{A_i\ \lambda_s\ G_s}_{\dot{Q}_{i\rightarrow crystal}} = \underbrace{A_i\ \lambda_L\ G_m}_{\dot{Q}_{men\rightarrow i}} + \underbrace{A_i\ \rho_s\ \Delta H_f\ v_g}_{\dot{Q}_{fusion}}. \tag{3.3}$$

The heat flow $\dot{Q}_{i\rightarrow crystal}$ from the interface into the crystal equals to the sum of the heat flow $\dot{Q}_{men\rightarrow i}$ from the meniscus into the interface and the heat of fusion released $\dot{Q}_{fusion}$. A_i is the cross section area of the interface, G_s and G_m are the temperature gradients along the crystal respectively the meniscus, λ_s and λ_L the thermal conductivities of solid respectively liquid silicon. Finally, ΔH_f is the specific latent heat of fusion and ρ_s the density of solid silicon. The expressions for the heat flows and for $\dot{Q}_{fusion}$ are derived in Section 3.2.2.
Finally equation (3.3) is solved for v_g.

Meniscus slope angle θ

The computation of the meniscus slope angle θ as a function of r and h, or, inverting the problem, of the meniscus height h as a function of r and θ, requires a detailed analysis of the meniscus surface curvature.
As stated in [8], the curvature of a non-planar contact surface between two fluids of different type (in the case of interest, the meniscus surface between molten silicon and the atmosphere in the puller) depends on the surface tension σ between these fluids and the capillary pressure Δp; given a point P on the surface, in this point the variables are related by the equation of Young-Laplace, which is a nonlinear partial differential equation (PDE):

$$\Delta p = \sigma \left(\frac{1}{R_1} + \frac{1}{R_2} \right).$$

The capillary pressure is the pressure variation which is encountered traversing the contact surface in P, and R_1 and R_2 are the local, principal radii of the curvature of the surface. Therefore, the two principal radii as well as the mean curvature, defined as $\frac{1}{2}\left(\frac{1}{R_1} + \frac{1}{R_2}\right)$, are subjected to a complex dependence from the pressure and the surface tension, in every point P of the contact surface.
For the PDE of Young-Laplace no exact closed-form solution is available; the only way to compute the meniscus curvature is therefore by numerical analysis. In the case of cylindrical symmetry, which is the case of interest, cylindrical coordinates can be introduced in the Young-Laplace equation, leading to a remarkably simplified expression, the so-called Euler-Laplace equation. Unfortunately, even in this latter case no exact closed-form solution can be derived. Chapter 8 of [8] provides a detailed analysis of these equations.
Anyway, in the case of interest it is sufficient to dispose of an expression which relates the meniscus height h to the meniscus slope angle θ and the crystal radius r. For this purpose, several closed-form approximations of the Euler-Laplace equation solution were proposed in literature over the last decades (see, for example, [3], [12] and [13]). The most popular approximations are:

- The Tsivinskii formula used in [1] for the original capillary model:

$$h = a \sqrt{1 - sin\theta + \frac{a^2\, cos^2\theta}{16\, r^2}} - \frac{a^2\, cos\theta}{4r}.$$

- The Boucher formula:

$$h = a \sqrt{\frac{1 - sin(\theta)}{1 + \frac{a}{\sqrt{2}\, r}}}.$$

- The Johansen-Boucher formula, which is an improved version of the Boucher formula, proposed by T. H. Johansen in [13]:

$$h = a\sqrt{\frac{1 - sin(\theta)}{1 + 0.6915\left(\frac{a}{r}\right)^{1.1}}}.$$

It is worth mentioning that, strictly speaking, the validity of these approximations is limited to the case $r \gg a$;[4] therefore, since the capillary constant for silicon is approximately 8 *mm*, the approximations hold only for the Body phase, where the crystal radius is of the order of hundreds of millimeters. However, as will be discussed below, it turns out that the formulae can be used also for the Neck phase.

From the comparison of the meniscus height computed by the three approximation formulae, reported in Figure 3.2 for a small radius (2.5 *mm*, i.e. Neck conditions) and in Figure 3.3 for a large radius (100 *mm*, i.e. Body conditions), it emerges that for small radii there is a remarkable difference between the formulae, whereas for large radii it is negligible. The circles in the figures identify the desired working point for crystal pulling with constant radius ($\theta = \theta_C \approx 11°$).

[12] reports a comparison of the Tsivinskii and the Boucher approximation with the exact numerical solution of the Euler-Laplace PDE, [13] compares the Boucher and the improved Boucher formula to the numerical solution (for small radii). From these considerations it is clear that the Tsivinskii formula offers the worst approximation, whereas the other formulae are close to the exact solution; as a matter of fact, the best approximation is yielded by the Johansen-Boucher formula.

Furthermore, from the comparisons reported by Johansen in [12] and [13], relating to small radii, it turns out that the formulae work fine also in this case, apart from the Tsivinskii approximation, which should therefore not be used for the Neck phase. In conclusion, the best result is given by using the Johansen-Boucher instead of the Tsivinskii formula applied in [1].

---

[4] as in [5], whereas the Young-Laplace and the Euler-Laplace equations are of general validity

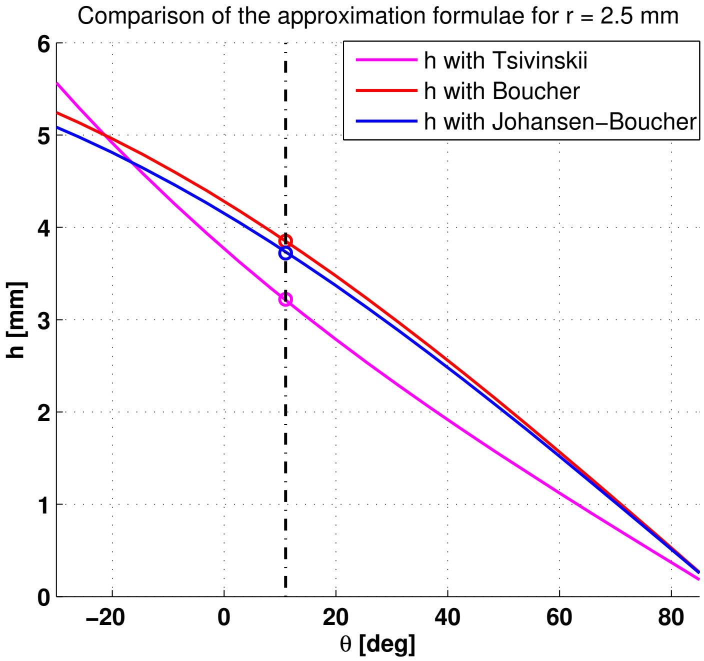

Figure 3.2: Comparison of the approximation formulae for $r = 2.5\ mm$

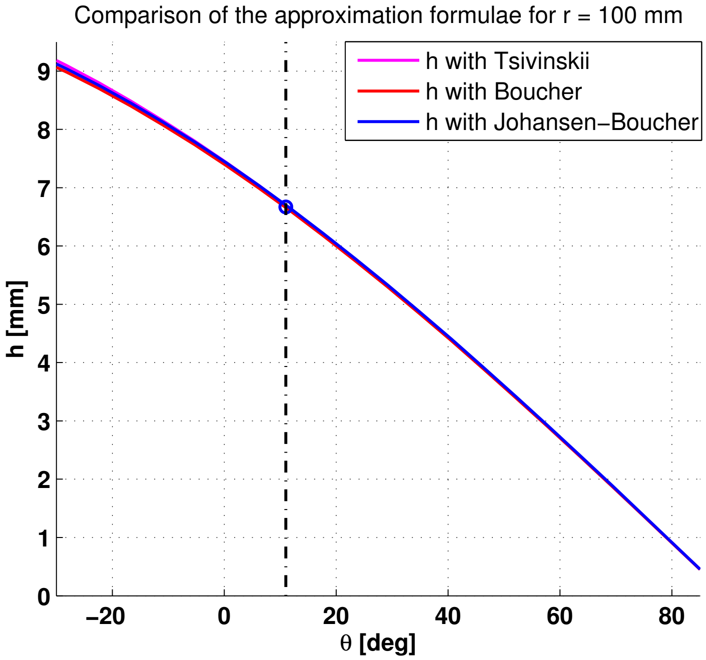

Figure 3.3: Comparison of the approximation formulae for $r = 100\ mm$

Meniscus height update equation

According to [3] the crystal growth rate v_g is related to the pull rate v by the following equation:

$$v_g = v - \dot{h} - v_{cruc} - \dot{h}_{melt}. \tag{3.4}$$

Here v_{cruc} is the vertical (i.e., along the z-axis) translation speed of the crucible and $\dot{h}_{melt}$ the variation of the height of the melt with respect to the crucible bottom. Obviously, $\dot{h}$ is the meniscus height derivative.
Therefore, as anticipated above, the dynamic of the growth rate does not equal the dynamic of pull rate,[5] since usually also the crucible is moving and, furthermore, the meniscus height and the melt level vary with time.

For neck pulling the melt level variations can be neglected, i.e. $\dot{h}_{melt} \approx 0$, and therefore v_{cruc} is set to zero; for the body phase the crucible lift rate is imposed to follow the level variation. For this purpose, $\dot{h}_{melt}$ can be computed explicitly by performing a mass balance at the interface. The overall mass of silicon contained in the puller, m_{tot}, is given by the sum of the overall mass of solid silicon contained in the crystal, m_{tot}, and the overall mass of liquid silicon; neglecting the mass contained in the meniscus, it is given by the mass of the melt, m_{melt}. Performing the derivative, keeping in mind that the overall mass m_{total} is constant, yields:

$$m_{total} = m_{crystal} + m_{melt} \quad \Longrightarrow \quad \dot{m}_{crystal} = -\dot{m}_{melt}.$$

The crystal and melt masses are, in a first approximation, assuming crystal and crucible of perfectly cylindrical shape, given by:

$$\begin{aligned} m_{crystal} &= A_i \, \rho_s \, l, \\ m_{melt} &= A_{cruc} \, \rho_m \, h_{melt}. \end{aligned}$$

Here $A_i = \pi \, r^2$ is the cross section area of the interface and $A_{cruc} = \pi \, R^2_{cruc}$ the area of the base of the crucible; l is the length of the crystal. Furthermore, ρ_s and ρ_m are the densities of solid respectively liquid silicon. Deriving these equations with respect to time yields the mass variations:

$$\begin{aligned} \dot{m}_{crystal} &= A_i \, \rho_s \, \frac{d\,l}{d\,t} = \pi \, r^2 \, \rho_s \, v_g, \\ \dot{m}_{melt} &= A_{cruc} \, \rho_m \frac{d\,h_m}{d\,t} = \pi \, R^2_{cruc} \, \rho_m \dot{h}_m. \end{aligned}$$

In fact, v_g is defined as the derivative of the crystal length. As mentioned above, for this mass balance the meniscus mass was neglected; therefore, for the balance, *and only for the balance*, the meniscus height variation $\dot{h}$ can

---

[5] only the average values are equal, as stated previously

be neglected too. For this reason in the equation for $\dot{m}_{crystal}$ the growth rate v_g can be approximated with the pull rate v. It is worthwhile to stress once more that this approximation can be used only in this particular case, whereas generally it does not hold, as discussed above.
Finally, substituting the mass variation formulae in the mass balance equation, solving it for $\dot{h}_m$ and setting $v_{cruc} = -\dot{h}_m$, the crucible lift rate set point is:

$$v_{cruc} = -\dot{h}_m \approx \frac{\rho_s\, r^2}{\rho_m\, R_{cruc}^2}\, v.$$

Melt temperature update equation

As explained at the beginning of this chapter, in absence of heater power variations the melt temperature, to be exact the melt temperature *under the meniscus*, is approximately constant. Actually, this temperature depends also on the crucible rotations and the magnetic field applied to the puller; anyway, in the conditions of interest (i.e. the Neck phase) these two parameters are kept constant and therefore the only significant dependence is the one from the heater power.
Therefore T_m should be modelled as follows:

$$T_m(n+1) = T_m(n) + c(P_{[0,\ n]}).$$

$c(\cdot)$ is a corrective function which is computed from the heater power applied in the past, $P_{[0,\ n]}$. However, it is impossible to establish a simple relation between temperature variations and heater power variations; for this reason, the effect of the heater power variations, always present since a second control loop is varying the power in order to adjust the growth rate, is mimicked with a random walk:

$$T_m(n+1) = T_m(n) + w_{T_m}(n).$$

w_{T_m} is a random process, more precisely a white noise process with zero mean and Gaussian probability distribution function.

3.2.2 Discussion of the model parameters

In this section some of the model parameters of Table 3.1 are discussed. Unfortunately, as will be pointed out below, most of these parameters are affected by huge uncertainties and therefore any equation which depends on one or more of them is fairly unreliable.
In the following, a brief description of the parameters is given:

- ***Melting point temperature of silicon*** T_c
  The melting point temperature of silicon depends, like for almost any other element, on the pressure. Usually a value referring to the standard atmospheric pressure, which is 1013.25 *mbar*, is used. However, the pressure of the atmosphere inside the puller is normally much lower, typically $50 \div 100$ *mbar*. The variation of T_c with respect to a certain pressure variation can be computed from the equation of Clausius-Clapeyron; as stated in [9], the temperature variation corresponding to a pressure variation of 1013.25 *mbar* amounts to 0.0034 *K* and is therefore negligible.

- ***Thermal conductivity of solid*** **(**λ_s**)** ***and liquid silicon*** **(**λ_L**)**
  The thermal conductivity λ is defined as the heat flow $\dot{Q}$ through a plate of area A subjected to the temperature gradient G [10]. It is the property of a material's ability to conduct heat; the lower λ, the better it insulates. From its definition

  $$\lambda = \frac{\dot{Q}}{A\,G},$$

  expression $\dot{Q} = \lambda\ A\ G$ follows for the heat flow, justifying the expressions used in the heat flow balance (3.3), required for the derivation of v_g.

- ***Density of solid*** **(**ρ_s**)** ***and liquid silicon*** **(**ρ_m**)** ***at temperature*** T_c
  Whereas the solid silicon density is known with quite good approximation, the value of the liquid silicon density is much more difficult to evaluate, and therefore many different values are proposed.

- ***Specific latent heat of fusion*** ΔH_f
  The latent heat Q associated with a thermodynamic transformation is the amount of energy released or absorbed by a body during a phase transition, without change of temperature (in the case of interest, fusion/freezing). Then, the specific latent heat ΔH is the ratio between Q and the transformed mass m:

  $$\Delta H = \frac{Q}{m}.$$

  From the definition of the specific latent heat it follows that the specific heat is given by

  $$Q = m\ \Delta H = \rho\ V\ \Delta H,$$

  where V is the volume and ρ the density of the transformed quantity. Now, lets consider the solidification of the liquid silicon on the interface between the meniscus and the crystal; from an energetically point of view, the latent heat assiocated with the solidification, Q_s, can equivalently be seen as (latent) heat of fusion Q_f. If the solidificated portion of the crystal is of perfectly cylindrical shape, the transformed volume is given by $V = A_i\, l$, where l is the length of the solidificated part and A_i the interface cross section area; finally, deriving with respect to time, the heat released by fusion, used in the heat flow balance (3.3), is obtained:

$$\dot{Q}_f = \frac{d\, Q_f}{d\, t} = \frac{d\, (\rho_s\, A_i\, l\, \Delta H_f)}{d\, t} = \rho_s\, A_i\, \Delta H_f\, \frac{d\, l}{d\, t} = \rho_s\, A_i\, \Delta H_f\, v_g.$$

Note that the definition of growth rate, $v_g = \frac{dl}{dt}$, was used in the last step.

- ***Capillary constant*** a
  The capillary constant for the Czochralski crystal growth process is defined as [3]

$$a = \sqrt{\frac{2\,\sigma}{\rho_m\, g}},$$

  where σ denotes the surface tension between the silicon melt and the atmosphere inside the puller and depends on the composition of the melt; g is the gravitational acceleration on Earth. It is worth mentioning that actually a is varying during the crystal pulling process, however in a first approximation it can be considered constant.

- ***Equilibrium growth angle*** θ_c
  The equilibrium growth angle is given by the difference between the meniscus slope angle θ and the crystal slope angle β; if the trijunction line is in thermodynamic equilibrium, θ_c is a material constant. Usually for silicon its value is situated between 11 and 12 degrees.
  Note that for $\theta = \theta_c$ the crystal slope angle is zero and therefore the crystal grows with a perfectly constant radius; this is the desired working condition in the Neck phase as well as in the Body phase.

- ***Temperature gradient over the crystal*** G_s
  The temperature gradient along the crystal is difficult to evaluate since it depends on many different parameters, especially on the configuration of the hot zone; for instance, the position of the reflector with respect to the melt surface and the dimensions of the graphite susceptor and the crucible are of importance. Furthermore, it varies with the pull rate, the actual crystal radius and the temperature on the surface of the crystal T_a; for a fixed hotzone it can be described as [4]:

$$G_s = [h_1(v) + h_2(r)]\ (T_c - T_a).$$

> Anyway, it is impossible to derive a sufficiently accurate closed-form expression for functions h_1 and h_2; typically the value of G_s is approximated with thermal simulations. For small variations of the crystal radius around a certain mean value G_s is approximately constant; this situation is typically verified in the Neck phase as well as in the Body phase.

From the parameter discussion it turns out that almost all model parameters, especially the gradient G_s, are affected by huge uncertainties. For this reason an improved version of the capillary model was developed, eliminating most of these parameters from the equations, as will be shown in the next section.

3.3 Description of the improved capillary state space model

As pointed out in the previous section, the original capillary model proposed in [1] suffers from some problems, namely:

- the use of the Tsivinskii formula for the meniscus slope angle computation; in the neck phase, where the crystal radius r is of the same order as the capillary constant a, this approximation of the numerical solution is not very precise;

- the expression used for v_g depends on many parameters affected by huge uncertainties and therefore the expression itself is unreliable.

Thus in the improved model the meniscus slope angle is computed inverting the Johansen-Boucher formula.
Regarding the second point, it can be stated that the modelling approach of [1] is that of disregarding the thermal model. This is reasonable, since it is extremely difficult to model the thermal behaviour of the crystal growth system with the required precision [3]; for this purpose, it would be necessary to resort to a spatially distributed model, which is far beyond the scope of this project. The only junction of the capillary model with the thermal part of the system is represented by the crystal growth rate; in fact, in its expression the melt temperature appears. However, from a geometrical point of view (the capillary model is a geometrical model) it is completely indifferent whether the thermal junction is observed through the growth rate or through the melt temperature; since the relation between the growth rate and the melt temperature is very unreliable, it would be far better to choose v_g for this purpose.
This approach yields a remarkable improvement of model robustness, since the number of model parameters is drastically reduced. Furthermore, the modified model is "more linear" than the original model; as a matter of fact, now also the second state variable h is computed as a perfectly linear function of the other states and the input variable $u = v$. The nonlinearity in the model is reduced to the first state equation. This is another important advantage, since the control approach (LQR) is a linear one.

3.4 Considerations on the linearized model

In this section a brief analysis of the *linearized* improved capillary model is performed. This analysis is indispensable for the design of the LQR law, described in Chapter 6. Linearizing in the operating point $(\bar{x},\ \bar{u})$ yields:[6]

$$\begin{aligned}[x(n+1) - f(\bar{x}, \bar{u})] &= F\,[x(n) - \bar{x}] + G\,[u(n) - \bar{u}], \\ y(n) &= H\,[x(n) - \bar{x}].\end{aligned}$$

The Jacobian matrices, evaluated in the operating point, are given by:

$$\begin{aligned}F &= \left.\frac{\partial\, f(x,u)}{\partial\, x}\right|_{(x,u)=(\bar{x},\ \bar{u})}, \\ G &= \left.\frac{\partial\, f(x,u)}{\partial\, u}\right|_{(x,u)=(\bar{x},\ \bar{u})}, \\ H &= \left.\frac{\partial\, h(x,u)}{\partial\, x}\right|_{(x,u)=(\bar{x},\ \bar{u})}.\end{aligned}$$

The system matrix F is:

$$F = \begin{bmatrix} \frac{\partial f_1}{\partial r} & \frac{\partial f_1}{\partial h} & \frac{\partial f_1}{\partial v_g} \\ \frac{\partial f_2}{\partial r} & \frac{\partial f_2}{\partial h} & \frac{\partial f_2}{\partial v_g} \\ \frac{\partial f_3}{\partial r} & \frac{\partial f_3}{\partial h} & \frac{\partial f_3}{\partial v_g} \end{bmatrix} = \begin{bmatrix} f_{11}(r,h) & f_{12}(r,h) & f_{13}(r,h) \\ 0 & 1 & -\Delta t \\ 0 & 0 & 1 \end{bmatrix}.$$

The exact expressions of the entries in the first row are not important; a brief, qualitative discussion of $f_{11}(r,h)$ is performed below.
Furthermore, matrix G is given by the simple expression:

$$G = \begin{bmatrix} \frac{\partial f_1}{\partial v} \\ \frac{\partial f_2}{\partial v} \\ \frac{\partial f_3}{\partial v} \end{bmatrix} = \begin{bmatrix} 0 \\ \Delta t \\ 0 \end{bmatrix}.$$

Finally, for the already linear output equation H is conserved:

$$H = [1 \quad 0 \quad 0].$$

From the system matrices it can be seen that only the system matrix F depends on the operating point, whereas G and H are constant. Furthermore, F is always an *upper triangular matrix* and therefore its eigenvalues are given by its diagonal entries:

$$(f_{11}(r,h),\ 1,\ 1).$$

---

[6] $f(\cdot)$ is defined as $[f_1(\cdot)\ \ f_2(\cdot)\ \ f_3(\cdot)]^T$

Since the state variables are always non-negative (as well as the parameters Δt and a), $f_{11}(r,h)$ is always of the form:

$$f_{11}(r,h) = 1 + \epsilon, \qquad \epsilon \geq 0.$$

However in the operating conditions of interest $\epsilon > 0$; that is, the linear system, as well as the nonlinear system, is unstable in the equilibrium points of interest (that is, constant radius pulling, with non-zero pull rate). It is worth mentioning that the same properties hold for the original capillary model.

Furthermore, for the LQR development the concepts of *controllability* and *stabilizability* of the linearized system are of interest (this will be pointed out in more detail in the LQR chapter). In order to verify the controllability, the controllability matrix is considered:

$$C = \begin{bmatrix} G & F\,G & F^2\,G \end{bmatrix} = \begin{bmatrix} 0 & * & * \\ * & * & * \\ 0 & 0 & 0 \end{bmatrix}.$$

The asterisks denote the non-zero entries; due to the particular upper triangular structure of F, the controllability matrix is always of this form, for each choice of the operating point. Since all entries of the third row are zero, the rank of C can not exceed 2, thus the linearized system is not controllable. Since the absolute value of all the eigenvalues is ≥ 1, furthermore there appears at least one non-stabilizable eigenvalue (more precisely, the third one) and therefore the system is not even stabilizable. Intuitively, this was obvious from the form of the state update equations: only r and h can be driven by the pullrate v, whereas it is not possible to act on v_g. Therefore, this state variable is not controllable, and the corresponding eigenvalue is not stabilizable. Again, this statement holds for the old capillary model too (however the non-controllable variable is T_m in that case).
As will be pointed out in the LQR chapter, the LQR law can not be applied to non-stabilizable systems; therefore an alternative system has to be considered in order to make things work out for the LQR design. This problem is addressed in Chapter 6.

Chapter 4

Statistical Filtering Approaches

4.1 Introduction

In almost any real system only a few variables of interest[1] are directly measurable. The remaining variables, the so-called inner state variables, can not be directly accessed. In the case of the capillary model, the only measured variable is the crystal radius, whereas the meniscus height and the growth rate are hidden variables, which can not be evaluated directly. The state estimation problem aims to the reconstruction of these inner state variables from the available measurements, relying on some kind of model which describes the relation among them. Furthermore, the estimator has to take into account the fact that the measurement is affected by measurement noise. This noise derives from the measurement device, in our case the camera.

The standard approach to the estimation problem is a statistical one; once a process model was developed, preferably in state space form, uncertainties and approximations introduced during modelling are described with zero mean white process noise, with an appropriately chosen covariance matrix. Also, measurement noise is introduced on the observation or measurement equation. Typically noises are considered to have Gaussian probability density function.[2]

In our case, the process model is given by the improved version of (3.1), discussed in Section 3.3.

The output (or measurement) equation is given by $y(n) = r(n) + w_y(n)$; w_y is the measurement noise ($y(n)$ is the measured radius, $r(n)$ the true radius). Furthermore process noise is added to the first and the second state equation, obtaining the model for the statistical filter/estimator.

---

[1] in our case, the variables of interest are the variables required for the LQR control

[2] anyway, the probability density function could be of any type

The process noise is defined as:

$$w_x = \begin{bmatrix} w_{x,1} \\ w_{x,2} \\ w_{x,3} \end{bmatrix}.$$

For this work, noises are modelled as Gaussian, wide-sense stationary white processes with zero mean:

$$w_x \sim \mathcal{N}(0,\ Q), \qquad w_y \sim \mathcal{N}(0,\ R).$$

Summarizing, we are in the case of a discrete-time nonlinear dynamic system, i.e. a system of the following form:

$$\begin{aligned} x(n+1) &= f(x(n), u(n), w_x(n)), \\ y(n) &= h(x(n), w_y(n)). \end{aligned} \tag{4.1}$$

x is the state vector of dimension n, u the input vector with dimension m (in our case $m = 1$, since the only input is the pull rate v) and y the output vector of dimension p. w_x has dimension n_x, w_y dimension n_y. In the following sections an overview on statistical filtering theory and the approaches used for this project is given, based on the above model. It is worth mentioning once more that the problem consists in the estimation of the state vector x, based on the measurement y and the process model. Note that no additivity assumption has been made for process and measurement noises, in order to describe the most general case.

4.2 The Bayesian estimation problem

Lets consider two random vectors, $\mathbf{X} \sim p_{\mathbf{X}}(X)$ of dimension n and $\mathbf{Y} \sim p_{\mathbf{Y}}(Y)$ of dimension m, related by the equation $\mathbf{Y} = h(\mathbf{X}, \mathbf{w})$. $h(\cdot)$ is an arbitrary, not necessary linear function, $\mathbf{Y}$ is the measured variable (by some measurement device), affected by measurement noise $\mathbf{w} \sim p_{\mathbf{w}}(w)$ (typically Gaussian distributed), whereas $\mathbf{X}$ is not accessible and has to be estimated from the observations Y of $\mathbf{Y}$.[3]

From a practical point of view, this probabilistic representation is absolutely reasonable, since the measurements delivered by a sensor are always affected by some degree of uncertainty. In absence of systematic measurement errors, the measurement noise will have zero mean.

In this paper the Bayesian approach to the estimation problem was adopted; the concepts described in this section are based on the treatment reported in [23]. Bayesian estimation is the best choice if some kind of probabilistic information, usually called "prior information", is available on the random vector $\mathbf{X}$. For most practical problems this is the case: for example, the nominal value of $\mathbf{X}$ could be known, together with the dispersion of its real values around the nominal value (the concepts of nominal value and dispersion can be readily translated in probabilistic concepts, i.e. mean and variance of a random vector). From a statistical point of view, the complete prior information is given by the pdf $p_{\mathbf{X}}(\cdot)$; usually it is known only partially, like in the example, where only the mean and the variance are available.

The aim of the Bayesian estimation approach is to compute the conditional pdf $p_{\mathbf{X}}(X|Y)$, i.e. the pdf of $\mathbf{X}$ given a sample of $\mathbf{Y}$. $p_{\mathbf{Y}}(Y|X)$ is always available from the (known) relation between $\mathbf{Y}$ and $\mathbf{X}$; therefore, if $p_{\mathbf{X}}(X)$ is completely known,[4] $p_{\mathbf{YX}}(Y, X)$ can be computed and, using the Bayesian rule, also $p_{\mathbf{X}}(X|Y)$:

$$p_{\mathbf{X}}(X|Y) = \frac{p_{\mathbf{YX}}(Y, X)}{p_{\mathbf{Y}}(Y)}. \tag{4.2}$$

$p_{\mathbf{X}}(X)$ is called the prior pdf, $p_{\mathbf{X}}(X|Y)$ the posterior pdf of $\mathbf{X}$. The posterior pdf represents the complete solution to the Bayesian estimation problem. Anyway, in most cases it is not possible to compute a closed-form description of the posterior pdf, and therefore typically only its first and second moment are estimated, i.e. its mean and covariance:

$$(m_{\mathbf{X}|\mathbf{Y}},\ P_{\mathbf{X}|\mathbf{Y}}).$$

---

[3] p denotes the probability density function, briefly pdf, of the random vector; furthermore, in this section the bold font is used to denote the random vector, whereas the normal font denotes an observation, or sample, of the random vector

[4] which usually is not the case

If the posterior pdf is Gaussian, the knowledge of the mean (denoted also with $\mathbb{E}[X|Y]$, where $\mathbb{E}$ is the expectation) and the covariance is equivalent to the knowledge of the complete pdf, given by

$$p_{\mathbf{X}}(X|Y) = \frac{1}{(2\pi)^{n/2}\sqrt{det(P_{\mathbf{X}|\mathbf{Y}})}}\, exp\left(-\frac{1}{2}(X - m_{\mathbf{X}|\mathbf{Y}})^T\, P^{-1}_{\mathbf{X}|\mathbf{Y}}\,(X - m_{\mathbf{X}|\mathbf{Y}})\right),$$

where n is the dimension of $\mathbf{X}$, *det* is the determinant of a square matrix and the superscript T denotes the transpose.

Returning to the inital problem of the estimation of $\mathbf{X}$, an estimator $\hat{\mathbf{X}}$ of $\mathbf{X}$ based on observations $\mathbf{Y}$ can be seen as a function $g(\mathbf{Y})$. In order to find the best estimation $g(\mathbf{Y})$ of $\mathbf{X}$, it is necessary to define a criterion or cost function which evaluates the quality of the estimation. In the context of Bayesian estimation, this criterion is the expected value of the square of the Euclidean distance between $\mathbf{X}$ and $g(\mathbf{Y})$; recalling the definition of the Euclidean distance,

$$d(\mathbf{X},\, g(\mathbf{Y})) = ||\mathbf{X} - g(\mathbf{Y})||_Q = \sqrt{(\mathbf{X} - g(\mathbf{Y}))^T\, Q\, (\mathbf{X} - g(\mathbf{Y}))},$$

where Q is a positive-definite square matrix of appropriate dimension, the cost function to minimize is:

$$c(\mathbf{X} - g(\mathbf{Y})) = \mathbb{E}\left[||\mathbf{X} - g(\mathbf{Y})||^2_Q\right]. \tag{4.3}$$

In these conditions, the following fundamental result holds [23]:

Theorem 1. *The conditional mean $\mathbb{E}[\boldsymbol{X}|\boldsymbol{Y}]$ is, amongst all functions $g : \mathbb{R}^m \to \mathbb{R}^n$ of the observation $\boldsymbol{Y}$, the one which minimizes the quadratic cost function (4.3), i.e.*

$$\mathbb{E}[\boldsymbol{X}|\boldsymbol{Y}] = Arg\, \underset{g(\cdot)}{min}\, \mathbb{E}\left[||\boldsymbol{X} - g(\boldsymbol{Y})||^2_Q\right],$$

for each positive-definite matrix $Q \in \mathbb{R}^{n\times n}$.♦

Since for the conditional mean it holds that $\mathbb{E}[\mathbb{E}[\mathbf{X}|\mathbf{Y}]] = \mathbb{E}[\mathbf{X}]$, it is an unbiased estimator[5] of $\mathbf{X}$, i.e. the estimation error $\mathbf{X} - \mathbb{E}[\mathbf{X}|\mathbf{Y}]$ has zero mean; therefore, choosing $Q = I$ (the identity matrix) the cost function for the conditional mean becomes the scalar variance of the estimation error:

$$\mathbb{E}\left[(\mathbf{X} - \mathbb{E}[\mathbf{X}|\mathbf{Y}])^T(\mathbf{X} - \mathbb{E}[\mathbf{X}|\mathbf{Y}])\right].$$

Therefore, from the theorem reported above it follows that the conditional mean minimizes the scalar estimation error variance! Summarizing, $\hat{\mathbf{X}} = \mathbb{E}[\mathbf{X}|\mathbf{Y}]$ is, amongst all unbiased estimators, the one with minimum estimation error variance; it is commonly known as ***minimum-variance unbiased***

---

[5] an estimator $\hat{\mathbf{X}}$ of random variable $\mathbf{X}$ is said to be unbiased if $\mathbb{E}[\hat{\mathbf{X}}] = \mathbb{E}[\mathbf{X}]$

estimator (MVUE).[6]
Unfortunately, in most practical cases the explicit computation of the conditional mean is not possible since it is a complex nonlinear function of the measurement. Anyway, there exists an important exception: when the vectors **X** and **Y** are jointly Gaussian distributed, the conditional mean becomes a linear function of the measurements [23]:

$$\begin{aligned} \mathbb{E}[\mathbf{X}|\mathbf{Y}] &= m_{\mathbf{X}} + P_{\mathbf{XY}}\, P_{\mathbf{Y}}^{-1}\, (\mathbf{Y} - m_{\mathbf{Y}}), \\ Var[\mathbf{X} - \mathbb{E}[\mathbf{X}|\mathbf{Y}]] &= P_{\mathbf{X}} - P_{\mathbf{XY}}\, P_{\mathbf{Y}}^{-1}\, P_{\mathbf{YX}}. \end{aligned}$$

The second expression gives the covariance matrix of the estimation error. $P_{\mathbf{X}}/P_{\mathbf{Y}}$ is the covariance matrix of **X** respectively **Y**, $P_{\mathbf{XY}}$ the cross covariance matrix between **X** and **Y** (furthermore, $P_{\mathbf{YX}} = P_{\mathbf{XY}}^T$); m denotes the means.

Anyway, as mentioned above, generally this is not the case. Since typically estimation problems have to be solved in real time, it is preferable to deal with linear or affine expressions; therefore, the class of admissible estimators is usually restricted to estimators which are linear/affine functions of **Y**. Then, the MVUE amongst these linear estimators is used. Obviously, in the Gaussian case the linear MVUE coincides with the MVUE in the general, nonlinear sense, as reported above.
Reporting only the result (its derivation is based on the Projection Theorem, for a detailed proof refer to [23]), the linear MVUE is given by:

$$\begin{aligned} \hat{\mathbf{X}}(\mathbf{Y}) &= m_{\mathbf{X}} + P_{\mathbf{XY}}\, P_{\mathbf{Y}}^{-1}\, (\mathbf{Y} - m_{\mathbf{Y}}), && (4.4) \\ Var[\mathbf{X} - \hat{\mathbf{X}}(\mathbf{Y})] &= P_{\mathbf{X}} - P_{\mathbf{XY}}\, P_{\mathbf{Y}}^{-1}\, P_{\mathbf{YX}}. && (4.5) \end{aligned}$$

These are exactly the same expressions as for the MVUE in the jointly Gaussian case. As will be shown in the next section, the Kalman Filter is just an efficient recursive algorithm which computes the quantities (4.4) and (4.5) in the case of a stochastic process described by a state space model.

This section is concluded with an important result, proved in [23], which will turn out to be useful in the following. Considering the linear MVUE defined by equations (4.4) and (4.5), the components of the measurement vector

$$\mathbf{Y} = \begin{bmatrix} \mathbf{Y}_1 \\ \vdots \\ \mathbf{Y}_m \end{bmatrix}.$$

can be considered as the observations relative to successive time steps, i.e. $\mathbf{Y}_1$ is the observation made at time step 1, $\mathbf{Y}_2$ the observation made at time

---

[6] or *optimal estimator in the mean-square sense* or *optimal minimum-variance estimator*

step 2 and so on. From this point of view, the estimator[7] $\hat{\mathbf{Y}}_t(\mathbf{Y}^{t-1})$ of component $\mathbf{Y}_t$, based on the observation of the vector

$$\mathbf{Y}^{t-1} = \begin{bmatrix} \mathbf{Y}_1 \\ \vdots \\ \mathbf{Y}_{t-1} \end{bmatrix},$$

can be interpreted as the one step ahead prediction of the measurements. Defining the measurement or output prediction error as

$$\mathbf{e}_t = \mathbf{Y}_t - \hat{\mathbf{Y}}_t(\mathbf{Y}^{t-1}), \qquad t = 1, 2, \ldots, m, \tag{4.6}$$

the following results holds:

Theorem 2. *The output prediction errors* $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots \boldsymbol{e}_m$*, defined by (4.6), are statistically orthogonal or uncorrelated, i.e.:*

$$\mathbb{E}[\boldsymbol{e}_t\, \boldsymbol{e}_s^T] = M\, \delta(t-s), \qquad t,\, s = 1, 2, \ldots, m.$$

Here, $M = M_t = \mathbb{E}[\boldsymbol{e}_t\, \boldsymbol{e}_t^T]$ *is the statistical power and* δ *the Kronecker delta function.*◆

Therefore the sequence of prediction errors, the so-called *innovation*, defines a white noise process (a formal definition of whiteness is given in Chapter 5).

4.3 The Kalman Filter for linear state space models

Lets return to the problem of state estimation in a state space model. If the model is linear, the standard, time-variant Kalman filter represents the linear MVUE derived in the previous section. In order to understand the principle of the Kalman filter extensions to the nonlinear case of our interest, it is worthwhile to take a look at the linear case. From now on, random variables and processes are represented with normal font too, in order to facilitate the readibility. Lets consider the linear equivalent to model (4.1),

$$\begin{aligned} x(n+1) &= F\, x(n) + G\, u(n) + w_x(n), \qquad x_0 = x(n_0), \\ y(n) &= H\, x(n) + w_y(n), \end{aligned} \tag{4.7}$$

with the following properties:[8]

---

[7] obtained setting $\mathbf{X} = \mathbf{Y}_t$ and $\mathbf{Y} = \mathbf{Y}^{t-1}$ in the linear MVUE equations

[8] here the typical case of zero mean, Gaussian distributed noises, Gaussian initial conditions and time-invariant matrices is presented; however, these assumptions can be relaxed. Kalman deduced the equations without any assumption on the probability distributions, furthermore the equations can be easily adjusted for noises with non-zero mean and for time-variant system matrices. However, the white noise assumption is vital and can not be relaxed.

- w_x is white Gaussian noise with:

$$E[w_x(n)] = 0, \qquad Var[w_x(n)] = Q, \qquad \forall n.$$

- w_y is white Gaussian noise with positive-definite covariance :

$$E[w_y(n)] = 0, \qquad Var[w_y(n)] = R > 0, \qquad \forall n.$$

- $E[w_x(n) w_y(m)^T] = S\,\delta(n-m)$.
- x_0 is a random vector which is uncorrelated with the process and measurement noises; its mean and variance are considered to be known:

$$m_0 = E[x_0], \qquad P_0 = Var[x_0].$$

The assumption $R > 0$ can be removed; if it does not hold, the pseudoinverse of R has to be considered in the filter equations instead of R^{-1}.

The Kalman filter is a recursive algorithm which solves the linear MVUE problem in this particular case; at every time step n, it computes the estimation of $x(n)$, based on the observations $Y^n = \{y(n_0+1), \ldots, y(n)\}$, by using a particular two-step structure: first, the filter computes the time update, i.e. the one step ahead prediction of $x(n)$ based on Y^{n-1}, and then the estimation is obtained from the prediction, corrected with the new measurement $y(n)$ (measurement update).
As a matter of fact, the Kalman filter is derived as the linear MVUE of the previous section, putting $\mathbf{X} = x(n)$ and

$$\mathbf{Y} = \begin{bmatrix} y(n) \\ \vdots \\ y(n_0+1) \end{bmatrix}.$$

A detailed deduction of the equations can be found, for example, in [23]; summarizing, the Kalman algorithm is structured as follows:

1. Definition of initial conditions for state prediction and state prediction error covariance as:

$$\hat{x}(n_0|n_0-1) = m_0, \qquad P(n_0|n_0-1) = P_0.$$

2. Computation of state prediction (time update), $n > n_0$:

$$\begin{aligned} \hat{x}(n|n-1) &= (F - S\,R^{-1}\,H)\,\hat{x}(n-1|n-1) + S\,R^{-1}\,y(n-1) + G\,u(n-1), \\ P(n|n-1) &= (F - S\,R^{-1}\,H)\,P(n-1|n-1)\,(F - S\,R^{-1}\,H)^T + (Q - S\,R^{-1}\,S^T). \end{aligned}$$

3. Computation of state estimation (measurement update):

$$\begin{aligned}
\hat{x}(n|n) &= \hat{x}(n|n-1) + L(n)\, e(n),\\
P(n|n) &= P(n|n-1) - P(n|n-1)\, H^T\, \Lambda(n)^{-1}\, H\, P(n|n-1).
\end{aligned}$$

The meaning of the variables is:

- $\hat{x}(n|n-1)$ is the prediction of state $x(n)$ based on measurements Y^{n-1};
- $\hat{x}(n|n)$ is the estimation of state $x(n)$ based on measurements Y^n; it is computed from the prediction $\hat{x}(n|n-1)$ by a correction factor depending on the *current* measurement $y(n)$;
- $e(n) = y(n) - H\, \hat{x}(n|n-1)$ is the output prediction error or innovation; it is defined as the difference between the current measurement $y(n)$ and its prediction based on the observations Y^{n-1} and represents the core of the correction factor mentioned at the previous point;
- $P(n|n-1)$ is the covariance matrix of the state prediction error, defined as $x(n) - \hat{x}(n|n-1)$;
- $P(n|n)$ is the covariance matrix of the state estimation error, defined as $x(n) - \hat{x}(n|n)$;
- $\Lambda(n)$ is the covariance matrix of the innovation process $e(n)$, given by:

  $$\Lambda(n) = H\, P(n|n-1)\, H^T + R.$$

- $L(n)$ is the filter gain and is defined as:

  $$L(n) = P(n|n-1)\, H^T\, \Lambda(n)^{-1}.$$

In most cases measurement noise is uncorrelated with process noise (i.e., $S = 0$) and therefore the equations simplify. Furthermore, they can be reorganized in order to improve numerical stability (for example, the Kalman filter in information form). Beside the time-variant filter there exists also a time-invariant version of the Kalman filter with constant filter gain, the so-called Steady-state Kalman Filter [23].

Finally, the Kalman Filter is asymptotically stable if and only if the pair $(F - S\, R^{-1}\, H,\ H)$ is detectable and the pair $(F - S\, R^{-1}\, H,\ C)$ is stabilizable, where C is a matrix square root of $Q - SR^{-1}S^T$.

Since the Kalman filter is the linear MVUE in the state space context, its properties are the same; if the distributions of process noise, measurement noise and initial state are Gaussian, the state vector and the measurement vector are jointly Gaussian and the Kalman estimation, in the generic case of non-Gaussian distributions only the linear MVUE, becomes the MVUE amongst the nonlinear estimators too.

Furthermore, the innovation process $e(n)$ plays the role of the innovation introduced at the end of the previous section; later on it will be shown that the whiteness condition is vital for the filter tuning, i.e. the adjustment of the covariances of the measurement and process noises.

4.4 The estimation problem for nonlinear systems

Returning to the nonlinear case, there exist two important extensions of the Kalman filter: the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF). Although the EKF represents the standard for nonlinear estimation problems, generally the UKF is known to yield better performance in most cases [24]. Like the Kalman filter, these two filters focus on the estimation of the conditional mean and its error covariance. Furthermore, a third approach is commonly used for nonlinear estimation: the Particle Filter (PF). It is a Monte Carlo approach to the filter problem and aims to the reconstruction of the entire conditional pdf $p_{\mathbf{X}}(X|Y)$ and not only its first and second moment. This method is really powerful, however its computational cost is by far higher. Therefore, it should be used only if EKF or UKF are not able to yield satisfying results or the entire posterior pdf is required.

Considering nonlinear system (4.1), where noises are not necessary Gaussian, the EKF approach approximates the pdf of the noises and the state vector with a Gaussian one, and operates, at each time n, a linearization of functions $f(\cdot)$ and $h(\cdot)$. As it is well known, a Gaussian random vector propagated through a linear transform preserves the Gaussianity (with modified mean and covariance). Based on this concept, the EKF applies the ordinary Kalman equations after linearization. However, this approach can introduce large errors in the transformed mean and covariance, and in the worst case this can lead to divergence of the filter [24].

The UKF approach approximates distributions instead of functions, using a deterministic sampling approach. Again, the state pdf is approximated with a Gaussian pdf, however the representation changes with respect to the EKF. Basically the UKF captures the true covariance and the true mean of the state distribution with a set of carefully chosen sigma point. These points are then propagated through the true nonlinear system dynamics, and the posterior distribution described by the transformed sigma points is again approximated with a Gaussian pdf. This operation is called *Unscented Transformation* and yields, in some cases, better performance than the EKF approach.

Finally, also the Particle Filter samples the prior pdf and propagates the samples through the nonlinear system dynamics. However, it is not a deterministic sampling as in the UT; the samples, or particles, are drawn from the pdf with a random approach. Therefore, a large number N_p of particles must be used. This increases the computational effort, but on the other hand for $N_p \to \infty$ the particles yield an exact representation of the posterior pdf, which is the most exhaustive solution to the Bayesian estimation problem, as explained previously; from this information, all moments of the pdf can be computed. Furthermore, the PF requires no assumptions on the involved pdfs; therefore, it can be applied to every type of nonlinearity and for multimodal pdfs.

4.5 The Extended Kalman Filter (EKF)

The following considerations are based on [23].

Consider a random vector y obtained by applying an affine transformation to a Gaussian distributed random vector $\xi \sim \mathcal{N}(m_\xi,\ P_\xi)$:[9]

$$y = A + B\,\xi.$$

It is well known that y is still a Gaussian distributed random vector, more precisely:

$$y \sim \mathcal{N}(A + B\,m_\xi,\ B\,P_\xi\,B^T). \tag{4.8}$$

However, for a generic nonlinear transformation $y = g(\xi)$ this is no longer true; generally, no assumptions can be made on the pdf of y; it could be even multimodal.

This is why the EKF linearizes the transformation locally; the algorithm is based on the following two steps (with ξ not necessarily Gaussian distributed):

1. Approximation of the true distribution p_ξ with a Gaussian pdf, characterized by the *true* mean and covariance:

$$\tilde{p}_\xi(\cdot) = \mathcal{N}(m_\xi,\ P_\xi).$$

2. Propagation of ξ through the linearized version of function $g(\cdot)$; more precisely, set $\tilde{y} = G\,\xi$, where G is the Jacobian Matrix of g, evaluated in m_ξ:

$$G = \left.\frac{\partial\, g(\xi)}{\partial\, \xi}\right|_{\xi = m_\xi}.$$

---

[9] A and B are constant matrices of appropriate dimensions

Then, the pdf of approximation $\tilde{y}$ is, according to (4.8):

$$\tilde{y} \sim p_{\tilde{y}}(\cdot) = \mathcal{N}(G\, m_{\xi},\ G\, P_{\xi}\, G^T).$$

Obviously $p_{\tilde{y}}(\cdot)$ is only an approximation of the true pdf $p_y(\cdot)$; generally, even the true mean and covariance of y differ from the values used in the approximation. If the nonlinearity is quite regular and the true pdfs are nearly Gaussian or at least unimodal, this principle yields good results. However, usually it is difficult to establish a priori whether these properties are satisfied or not. If the pdfs are multimodal (Figure 4.1 reports an unimodal and a multimodal pdf; obviously, the Gaussian pdf is unimodal), the EKF approach is completely unappropriate.

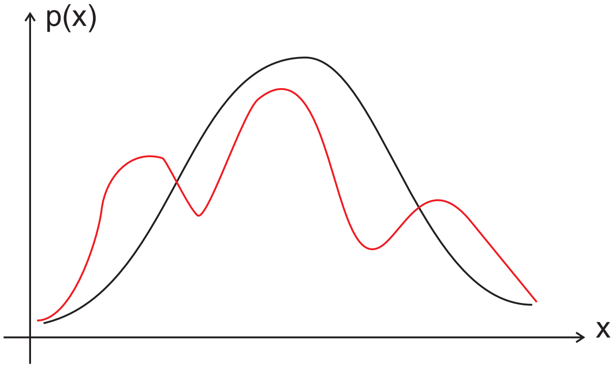

Figure 4.1: Unimodal pdf (black) vs. multimodal pdf (red)

The EKF applies the “Gaussian“ linearization idea to the equations of the nonlinear state space model (4.1):

$$\begin{aligned} x(n+1) &= f(x(n), u(n), w_x(n)), \\ y(n) &= h(x(n), w_y(n)). \end{aligned}$$

Then the standard Kalman filter equations reported previously are applied to the linearized system.
More precisely, at each time n the state function f is linearized around the best estimation of the vector

$$(x(n-1),\ u(n-1),\ w_x(n-1)),$$

which in this context is represented by the means:[10]

$$p_f = (\hat{x}(n-1|n-1),\ u(n-1),\ 0).$$

$\hat{x}(n-1|n-1)$, the estimation computed by the EKF at the previous step, is the best available estimation of the mean of $x(n-1)$.

---

[10] as for the Kalman equations, noises are supposed to have zero mean

Furthermore, the output function h is linearized around the best estimation of the mean of:

$$(x(n),\ w_y(n)).$$

Since the estimation $\hat{x}(n|n)$ is not yet available, the prediction $\hat{x}(n|n-1)$ is used. Therefore:

$$p_h = (\hat{x}(n|n-1),\ 0).$$

Linearizing model (4.1) this way yields:

$$\begin{aligned} x(n) &= f(p_f) + F_{n-1|n-1}\,[x(n-1) - \hat{x}(n-1|n-1)] + \ldots \\ \ldots & \quad +G_{n-1|n-1}\,u(n-1) + M_{n-1|n-1}\,w_x(n-1), \\ \\ y(n) &= h(p_h) + H_{n|n-1}\,[x(n) - \hat{x}(n|n-1)] + N_{n|n-1}\,w_y(n). \end{aligned}$$

The Jacobian matrices are evaluated in p_f respectively p_h (note that F, G and H are the matrices defined in Section 3.4):

$$\begin{aligned} F_{n-1|n-1} &= \left.\frac{\partial\, f(x,u,w_x)}{\partial\, x}\right|_{p_f}, \\ G_{n-1|n-1} &= \left.\frac{\partial\, f(x,u,w_x)}{\partial\, u}\right|_{p_f}, \\ M_{n-1|n-1} &= \left.\frac{\partial\, f(x,u,w_x)}{\partial\, w_x}\right|_{p_f}, \\ H_{n|n-1} &= \left.\frac{\partial\, h(x,w_y)}{\partial\, x}\right|_{p_h}, \\ N_{n|n-1} &= \left.\frac{\partial\, h(x,w_y)}{\partial\, w_y}\right|_{p_h}. \end{aligned}$$

The linearized model is of the same form as the linear system (4.7) and therefore the equations of the standard Kalman filter can be applied to it, with some small adjustments. The final EKF algorithm is reported below for the typical case of $S = 0$ (i.e., process noise and measurement noise are uncorrelated):

1. Initial conditions for state prediction and state prediction error variance:

$$\hat{x}(n_0|n_0-1) = m_0, \qquad P(n_0|n_0-1) = P_0.$$

2. State prediction (time update), $n > n_0$:

$$\begin{aligned} \hat{x}(n|n-1) &= f(\hat{x}(n-1|n-1),\ u(n-1), 0), \\ P(n|n-1) &= F_{n-1|n-1}\,P(n-1|n-1)\,F^T_{n-1|n-1} + M_{n-1|n-1}\,Q\,M^T_{n-1|n-1}. \end{aligned}$$

3. State estimation (measurement update):

$$\begin{aligned}\hat{x}(n|n) &= \hat{x}(n|n-1) + L(n)\ [y(n) - h(\hat{x}(n|n-1),\ 0)],\\ P(n|n) &= P(n|n-1) - P(n|n-1)\ H^T_{n|n-1}\ \Lambda(n)^{-1}\ H_{n|n-1}\ P(n|n-1).\end{aligned}$$

With:

- *Approximated* covariance matrix of the pseudo-innovation $e(n) = y(n) - h(\hat{x}(n|n-1),\ 0)$:

  $$\Lambda(n) = H_{n|n-1}\ P(n|n-1)\ H^T_{n|n-1} + N_{n|n-1}\ R\ N^T_{n|n-1}.$$

  In the nonlinear context it is not appropriate to call $e(n)$ innovation, since it is no longer rigorously white noise; therefore, the term “pseudo“ is used [23].

- Filter gain:

  $$L(n) = P(n|n-1)\ H^T_{n|n-1}\ \Lambda(n)^{-1}.$$

With respect to the linear case, the following modifications were applied:

- The variances of process and measurement noise are changed:

  $$\begin{aligned} Q &\longrightarrow M_{n-1|n-1}\ Q\ M^T_{n-1|n-1}\,, \\ R &\longrightarrow N_{n-1|n-1}\ R\ N^T_{n-1|n-1}\,. \end{aligned}$$

  This is due to the fact that in the linearized system the process and measurement noises are given by $M\ w_x$ respectively $N\ w_y$.

- The predictions $\hat{x}(n|n-1)$ and $\hat{y}(n|n-1)$ are calculated on the initial, nonlinear system, in order to reduce the approximation errors due to the linearization. The linearization is used only for the computation of the covariance matrices and for the filter gain. With this intelligent linearization approach the linearization noise can be kept typically at an acceptable level [23].

As stated in [23], the EKF presents some disadvantages with respect to the linear case. In particular, since for the prediction calculation the nonlinear equations are used, the EKF predictor as well as the EKF estimator are *nonlinear* functions of the measured variables and therefore of difficult statistical interpretation: nothing can be said, a priori, about the probability distributions.
Furthermore, the EKF has a remarkable computational cost with respect to the standard Kalman filter, since at every step the Jacobian matrices have to be calculated; for state estimation, it is of the order of $\mathcal{O}(n^3)$, where n is

the state vector dimension [35]. Anyway, for low order models as in our case, this is not relevant.

As stated at the beginning of this section, generally it is difficult to establish a priori whether the EKF can yield reasonable results or not, given a particular problem. Its performance depends heavily on the form of the nonlinearities and on the true state (and noise) pdfs, which usually are unknown. In any case, the EKF estimation is neither unbiased nor consistent; as a matter of fact it typically underestimates the estimation error covariances [23]. Furthermore, it is very sensible to the initial conditions (often, a wrong choice leads to divergence).
The Unscented Kalman Filter, introduced in the next section and already implemented in [1] with the original capillary model, can yield a better performance from certain points of view; however it is limited to unimodal pdfs too.

4.6 The Unscented Kalman Filter (UKF)

4.6.1 The Unscented Transformation (UT)

The Unscented Transformation is based on the so-called *sigma points*. Many different choices for these points are proposed in literature ([24], [25]): typically $2N + 1$ sigma points are used, where N is the dimension of the augmented state space (introduced in the next section). The set implemented for this project is described in the following; for now, the principles of the Unscented Transformation are pointed out without any specific considerations on sigma points and weights. The treatment is based on [27].

Reconsider the random vector ξ, with true mean m_ξ and true covariance matrix P_ξ, and the nonlinear transformed vector $y = g(\xi)$ of the previous section. Like for the EKF, the true pdf of ξ is approximated with a Gaussian:

$$\tilde{p}_\xi(\cdot) = \mathcal{N}(m_\xi, P_\xi). \tag{4.9}$$

The Unscented Transformation approximates the density $\tilde{p}_\xi(\cdot)$ with a set of carefully chosen sigma points, maps them through the nonlinear function g, and approximates the transformed sigma points again with a Gaussian pdf. The sigma points have to be chosen in order to capture m_ξ and P_ξ exactly; furthermore, they should capture, as good as possible, the true mean and covariance of y.

For the random vector $\xi \in \mathbb{R}^N$ (with positive-definite covariance matrix) the set $\Sigma = \{x \in \mathbb{R}^N \mid (x - m_\xi)^T \, P_\xi^{-1} \, (x - m_\xi) = 1\}$ defines an ellipse of dimension N centered in m_ξ. Its axes are given by the eigenvectors of

P_ξ, and the length over these axes by the square root of the corresponding eigenvalue [26]. More precisely, consider the Eigenvalue Decomposition $P_\xi = U_\xi\, \Sigma_\xi\, U_\xi^T$ of the covariance matrix (that is, a particular case of the Singular Value Decomposition, briefly SVD):

$$P_\xi = \begin{bmatrix} u_{1,\xi}\; u_{2,\xi} \dots u_{N,\xi} \end{bmatrix} \begin{bmatrix} \lambda_{1,\xi} & 0 & \dots & 0 \\ 0 & \lambda_{2,\xi} & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_{N,\xi} \end{bmatrix} \begin{bmatrix} u_{1,\xi}^T \\ u_{2,\xi}^T \\ \vdots \\ u_{N,\xi}^T \end{bmatrix}. \tag{4.10}$$

$\lambda_{i,\xi}$ are the eigenvalues and $u_{i,\xi}$ the corresponding eigenvectors; then the axes of the ellipse are given by $u_{i,\xi}$ and the corresponding length by $\sqrt{\lambda_{i,\xi}}$. Therefore, every (positive-definite) covariance matrix can be represented graphically with the associated ellipse, as shown in Figure 4.2 for $N = 2$.

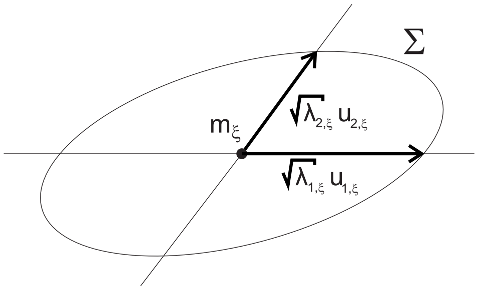

Figure 4.2: Ellipse Σ associated to covariance matrix P_ξ for $N = 2$

The UKF approach approximates this geometrical representation of the variance with a set of $L = 2N + 1$ carefully chosen points distributed around m_ξ; more precisely:[11]

$$\begin{aligned} \mathcal{P}_0 &= m_\xi, \\ \mathcal{P}_{1/2} &= m_\xi \pm \alpha\sigma_1, \\ &\vdots \\ \mathcal{P}_{2N-1/2N} &= m_\xi \pm \alpha\sigma_N. \end{aligned}$$

α is used as a scaling factor in order to determine the spread of the sigma points around the mean, and the vectors σ_i, $i = 1, \dots N$, are the columns

---

[11] the following considerations refer to a standard set of $2N + 1$ points; however, there exist also different choices

of a square root of the covariance matrix, i.e.:

$$P_\xi = \begin{bmatrix} \sigma_1 \, \sigma_2 \ldots \sigma_N \end{bmatrix} \begin{bmatrix} \sigma_1^T \\ \sigma_2^T \\ \vdots \\ \sigma_N^T \end{bmatrix} = \sum_{i=1}^{N} \sigma_i \, \sigma_i^T.$$

From a comparison with the Eigenvalue Decomposition (4.10) it is obvious that one possible choice is:

$$\sigma_i = \sqrt{\lambda_i} \, u_{i,\xi}, \qquad i = 1, \ldots, N.$$

Another possibility is to use the columns of the Cholesky factorization of the covariance matrix [35].
These points are then used to approximate the probability density (4.9) with the discrete pdf

$$\tilde{p}_\xi(\xi) = \sum_{i=0}^{2N} w_i \, \delta(\xi - \mathcal{P}_i) \approx p_\xi(\xi),$$

i.e. with the sum of $2N + 1$ sigma points centered in $\mathcal{P}_i$ and weighted with w_i, $i = 0, \ldots, 2N$. Since we are approximating a pdf, the weights have to be normalized, that is $\sum_{i=0}^{2N} w_i = 1$. For the rest, the only constraint is to choose them in order to have

$$\begin{aligned} \mathbb{E}_{\tilde{p}_\xi}[\xi] &= m_\xi, \\ Var_{\tilde{p}_\xi}[\xi] &= P_\xi, \end{aligned}$$

i.e. in order to capture the true mean and the true covariance of ξ.

Now, since we are interested in the pdf of the output variable $y = g(\xi)$, the points $\mathcal{P}_i$ are mapped through the nonlinear function g, obtaining the transformed sigma points:

$$\mathcal{Y}_0 = g(\mathcal{P}_0), \quad \mathcal{Y}_1 = g(\mathcal{P}_1), \quad \ldots, \quad \mathcal{Y}_{2N} = g(\mathcal{P}_{2N}).$$

The situation is represented in Figure 4.3, for $N = 2$ (and $\alpha = 1$). Note that m_y is the true mean of y and Σ_y the ellipse associated to the true covariance matrix of y, i.e P_y.
The transformed points are then used to approximate the true pdf of y:

$$\tilde{p}_y(y) = \sum_{i=0}^{2N} w_i \, \delta(y - \mathcal{Y}_i) \approx p_y(y).$$

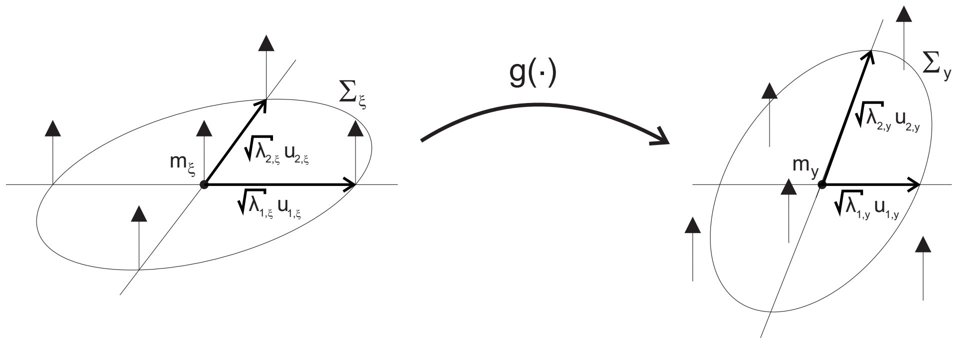

Figure 4.3: Unscented Transformation for $N = 2$ ($\alpha = 1$)

This choice yields for the mean and covariance (see [27] for the simple proof):

$$\tilde{m}_y = \mathbb{E}_{\tilde{p}_y}[y] = \sum_{i=0}^{2N} w_i\, \mathcal{Y}_i\,, \tag{4.11}$$

$$\tilde{P}_y = Var_{\tilde{p}_y}[y] = \sum_{i=0}^{2N} w_i\, (\mathcal{Y}_i - \tilde{m}_y)\, (\mathcal{Y}_i - \tilde{m}_y)^T. \tag{4.12}$$

Finally, the probability density of y is approximated with a Gaussian one, with mean and covariance matrix as given from the sigma point distribution:

$$\tilde{\tilde{p}}_y(\cdot) = \mathcal{N}(\tilde{m}_y,\ \tilde{P}_y) \approx \tilde{p}_y(\cdot) \approx p_y(\cdot).$$

The quality of the approximation $\tilde{\tilde{p}}_y(\cdot) \approx p_y(\cdot)$ depends heavily on the choice of the sigma point set. Anyway, as will be pointed out in the next section, only the mean and the covariance are of interest; therefore, it is sufficient to design the points in order to capture, as good as possible, these two moments of the pdf.

4.6.2 Application of the UT to the system equations

Reconsider the nonlinear state space model (4.1):

$$\begin{aligned} x(n+1) &= f(x(n), u(n), w_x(n)), \\ y(n) &= h(x(n), w_y(n)), \end{aligned}$$

with $x \in \mathbb{R}^n$, $w_x \in \mathbb{R}^{n_x}$ and $w_y \in \mathbb{R}^{n_y}$.

The following discussion, based again on [27], can be simplified in the particular case of purely additive noises (as in the capillary model), more precisely it is not necessary to augment the state vector. However, here the augmented state approach is presented, since it is more general and, for additive noises, equivalent to the non-augmented state space formulation.

In order to derive the Unscented Kalman Filter equations it is necessary to apply the Unscented Transformation to the system (4.1); the UKF structure is still recursive, that is the following operations have to be applied at each time n. Lets consider the augmented state vector:

$$x^a = \begin{bmatrix} x \\ w_x \\ w_y \end{bmatrix} \in \mathbb{R}^N, \qquad N = n + n_x + n_y.$$

Therefore at each time n, with reference to the previous section, $x^a(n-1)$ is the random vector ξ, the functions $f(\cdot)$ and $h(\cdot)$ play the role of $g(\cdot)$, and finally $x(n)$ and $y(n)$ play the role of the output random variable y.
The nonlinear system in terms of the augmented state can be rewritten as:

$$\begin{aligned} x(n+1) &= f(x^a(n),\, u(n)), \\ y(n) &= h(x^a(n)). \end{aligned}$$

$f(\cdot)$ and $h(\cdot)$ are still the same functions, this is just a different way to write the equations.
Now, according to the theory of the Unscented Transformation the - not necessary Gaussian - noise probability densities are first approximated as:[12]

$$\begin{aligned} p_{w_x(n)} &\approx \mathcal{N}(0,\, Q), \\ p_{w_y(n)} &\approx \mathcal{N}(0,\, R). \end{aligned}$$

Furthermore, the pdf of the random vector $x(n)$ is approximated as Gaussian too:

$$p_{x(n)} \approx \mathcal{N}(\hat{x}(n|n),\, P(n|n)).$$

Clearly, since the true mean and covariance matrix of $x(n)$ are not known, an approximation is used for these two parameters: like for the standard Kalman filter $\hat{x}(n|n)$ is the state estimation and $P(n|n)$ the state estimation error covariance at time n. Finally, state vector, process noise and measurement noise are supposed to be mutually uncorrelated:

$$x(n) \perp w_x(n) \perp w_y(n).$$

---

[12] again, noises are supposed to have zero mean; Q and R are the true covariance matrices

In these conditions, the pdf of the augmented state vector $x^a(n)$ is approximated as:

$$x^a(n) = \begin{bmatrix} x(n) \\ w_x(n) \\ w_y(n) \end{bmatrix} \implies \tilde{p}_{x^a(n)}(\cdot) = \mathcal{N}\left(\begin{bmatrix} \hat{x}(n|n) \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} P(n|n) & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & R \end{bmatrix}\right).$$

Now the sigma point set is generated, as shown in the previous section, with respect to the probability density $\tilde{p}_{x^a(n)}(\cdot)$, i.e. with respect to its mean and covariance matrix, denoted with $\hat{x}^a(n|n)$ and $P^a(n|n)$ from now on. Note that these sigma points are N-dimensional and not n-dimensional, in fact

$$\mathcal{X}_i^a = \begin{bmatrix} \mathcal{X}_i \\ \mathcal{W}_{\mathcal{X},i} \\ \mathcal{W}_{\mathcal{Y},i} \end{bmatrix} \in \mathbb{R}^N, \quad \text{with} \quad \mathcal{X}_i \in \mathbb{R}^n, \quad \mathcal{W}_{\mathcal{X},i} \in \mathbb{R}^{n_x}, \quad \mathcal{W}_{\mathcal{Y},i} \in \mathbb{R}^{n_y},$$

for $i = 0, \ldots, 2N$. In the following the notations $\mathcal{X}_i(n-1)$, $\mathcal{W}_{\mathcal{X},i}(n-1))$ and $\mathcal{W}_{\mathcal{Y},i}(n-1))$ are used for the sigma points computed from the augmented state $\hat{x}^a(n-1|n-1)$ and $P^a(n-1|n-1)$.

Summarizing, the UKF algorithm is structured as follows [24]:

1. The UKF is initialized with reference to the estimation:[13]

$$\hat{x}^a(n_0|n_0) = \begin{bmatrix} m_0 \\ 0 \\ 0 \end{bmatrix}, \qquad P(n_0|n_0) = \begin{bmatrix} P_0 & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & R \end{bmatrix}.$$

   Then, for $n > n_0$, the following steps are repeated.

2. The sigma points $\mathcal{X}_i(n-1)$, $\mathcal{W}_{\mathcal{X},i}(n-1)$ and $\mathcal{W}_{\mathcal{Y},i}(n-1)$, $i = 0, \ldots, 2N$, are computed from $\hat{x}^a(n-1|n-1)$ and $P^a(n-1|n-1)$.

---

[13] whereas KF and EKF are initialized with respect to the prediction

3. State prediction (time update):

$$\mathcal{X}_i(n|n-1) = f(\mathcal{X}_i(n-1),\, u(n-1),\, \mathcal{W}_{\mathcal{X},i}(n-1)), \quad i = 0, \ldots, 2N,$$

$$\hat{x}(n|n-1) = \sum_{i=0}^{2N} w_i\, \mathcal{X}_i(n|n-1),$$

$$P(n|n-1) = \sum_{i=0}^{2N} w_i\, [\mathcal{X}_i(n|n-1) - \hat{x}(n|n-1)]\, [\mathcal{X}_i(n|n-1) - \hat{x}(n|n-1)]^T.$$

The time update refers to the sigma points, i.e. the nonlinear state equation is applied to the sigma points derived from the previous state estimation, according to the principle of the UT.
$\hat{x}(n|n-1)$ is defined as the conditional mean $\mathbb{E}[x_n|Y^{n-1}]$; the mean is computed with respect to the discrete pdf described by the sigma points, and therefore it is given by expression (4.11), whereas the state prediction error covariance $P(n|n-1)$ is given by (4.12).
Output prediction $\hat{y}(n|n-1)$ and pseudo-innovation[14] covariance $\Lambda(n)$ are computed as follows:

$$\mathcal{Y}_i(n|n-1) = h(\mathcal{X}_i(n|n-1),\, \mathcal{W}_{\mathcal{Y},i}(n-1)), \qquad i = 0, \ldots, 2N,$$

$$\hat{y}(n|n-1) = \sum_{i=0}^{2N} w_i\, \mathcal{Y}_i(n|n-1),$$

$$\Lambda(n) = \sum_{i=0}^{2N} w_i\, [\mathcal{Y}_i(n|n-1) - \hat{y}(n|n-1)]\, [\mathcal{Y}_i(n|n-1) - \hat{y}(n|n-1)]^T.$$

Cross covariance between state prediction and pseudo-innovation:

$$P^{xe}(n|n-1) \quad = \quad \sum_{i=0}^{2N} w_i\, [\mathcal{X}_i(n|n-1) - \hat{x}(n|n-1)]\, [\mathcal{Y}_i(n|n-1) - \hat{y}(n|n-1)]^T.$$

4. State estimation (measurement update):

$$\hat{x}(n|n) = \hat{x}(n|n-1) + P^{xe}(n|n-1)\, \Lambda(n)^{-1}\, [y(n) - \hat{y}(n|n-1)],$$

$$P(n|n) = P(n|n-1) - P^{xe}(n|n-1)\, \Lambda(n)^{-1}\, P^{xe}(n|n-1)^T.$$

Therefore, the UKF preserves the typical time update-measurement update structure of the standard Kalman filter. The meaning of the the variables is analogue to the KF and the EKF.
The UKF discussion is concluded with the important point of the choice of the sigma point set.

---

[14] as for the EKF, it is no longer appropriate to call the output prediction error innovation

4.6.3 The sigma point set

For this project the set implemented for the UKF in [1] was used; it is reported, for instance, in [24]. From a literature review (amongst others, papers [24], [25] and [35] were considered) it results that this set is one of the most complete sets developed so far; many other sets with a reduced number of sigma points are proposed, anyway they are interesting only from the point of view of the computational cost. Furthermore, during the analysis of the capillary system it turned out (this will be discussed further on) that the involved state variables have unimodal, almost Gaussian distribution; for this reason, no other sets were tested, since in this case it is not possible to significantly improve the filter performance with a different choice of sigma points.
The set used in [1] consists of the following $(2N+1)$ sigma points centered in the current (augmented) state estimation $\hat{x}^a(n|n)$ (the time dependency was omitted in order to simplify the notation):[15]

$$\begin{aligned} \mathcal{X}_0^a &= \hat{x}^a, \\ \mathcal{X}_i^a &= \hat{x}^a \pm \alpha\sqrt{N+k}\,\sigma_i, \qquad i = 1,\ldots,N, \end{aligned}$$

where σ_i, $i = 1,\ldots N$, are the columns of the Cholesky factorization of P^a, that is the covariance matrix of the current augmented state estimation error:

$$P^a = Q_{chol}Q_{chol}^T = [\sigma_1\ \ldots\sigma_N]\begin{bmatrix}\sigma_1^T \\ \vdots \\ \sigma_N^T\end{bmatrix}.$$

As a matter of fact, typically the Cholesky factorization is used, since its computational cost is lower with respect to the Eigenvalue Decomposition. Finally, the weights of the sigma points are:

$$\begin{aligned} W_0^m &= \frac{N(\alpha^2-1)+\alpha^2\,k}{\alpha^2\,(N+k)}, \\ W_0^c &= \frac{N(\alpha^2-1)+\alpha^2\,k}{\alpha^2\,(N+k)} + 1 - \alpha^2 + \beta, \\ W_i^m &= W_i^c = \frac{1}{2}\frac{1}{\alpha^2\,(N+k)}, \qquad i = 1,\ldots,2N. \end{aligned}$$

The superscript "m" indicates the weights used in the state prediction calculation for the time update of the filter and the superscript "c" the weights used for the covariance calculations in the time and measurement update

---

[15] N is the dimension of the augmented state

equations. These weights differ only for $i = 0$.
According to [24], α determines the spread of the sigma points around $\hat{x}^a$; typically it is set to a small positive value like 10^{-3}. In some cases changing α yields no significant effect on the filter performance; typically, this happens when the nonlinearities are quite regular, as in our case. k is a secondary scaling parameter usually set to 0 and β is used to take into account prior knowledge on the involved pdfs; for almost Gaussian distributions, as in our case, it is set to 2. Obviously, in the case of the UKF the filter tuning consists not only in the tuning of the measurement and process noise covariances, but also in the choice of these sigma point parameters.

4.7 EKF vs. UKF: a comparison

Before proceeding to the Particle Filter exposition, a brief comparison between the EKF and the UKF approach is given. Lets consider again the nonlinear relation $y = g(\xi)$, introduced previously; according to [36] the central moments of the pdf of y are functions of the central moments of the pdf of ξ. More precisely, considering only the mean and the covariance of y, it holds that:

$$\begin{aligned} m_y &= g_1(m_\xi = \mathbb{E}[\xi],\, P_\xi = \mathbb{E}[(\xi - m_\xi)\,(\xi - m_\xi)^T],\, \ldots), \\ P_y &= g_2(m_\xi,\, P_\xi,\, \ldots). \end{aligned}$$

For the following qualitative considerations, the exact expressions of $g_1(\cdot)$ and $g_2(\cdot)$ are not required; for a detailed analysis refer to [25]. It can be stated that [36]:

- obviously, according to its definition, the EKF catches m_ξ and P_ξ exactly; the same holds for the UKF, if a normalized set of weights is considered. The following points refer to this condition;
- the EKF approximation of m_y is correct up to the first order[16] in terms of the central moments of ξ, whereas the UKF approximation is correct up to the *second* order;
- both the EKF and UKF approximations of P_y are correct up to the second order in terms of the central moments of ξ;
- in the case of the sigma point set considered in the previous section, adjusting α and β it is possible to improve the approximation with respect to the third and fourth order moments of ξ;
- if the true pdf of ξ is symmetric, then the UKF approximations of m_y and P_y are correct up to the third order (since the third order central moment of a symmetric pdf is zero);

---

[16] here, the formulation "up to the nth order" includes the nth order

- the computational cost is $\mathcal{O}(n^3)$ for both filters, where n is the state space dimension (if the augmented UKF is considered, then $n \to N$).

From these considerations it turns out that in certain cases the UKF approximation of m_y and P_y is more precise then the approximation yielded by the EKF. However, in many practical cases, namely when the pdfs are almost Gaussian, since the nonlinearities are quite smooth, these improvements introduced by the UKF are not significant. As will be shown later, this is the case for the capillary model. Furthermore, the UKF tuning is less immediate than the EKF tuning, since also the sigma point parameters have to be adjusted.

Finally, both filters yield reasonable results only in the case of unimodal pdfs. This problem is addressed by the Particle Filter, which can be applied to every type of nonlinearity and pdf, and furthermore yields the *exact and complete* solution to the Bayesian estimation problem, for a large number of particles. However, its computational cost is by far higher. In this project it was considered only in order to verify the Gaussian pdf hypothesis on the state variables of the capillary model.

4.8 The Particle Filter (PF)

The exposition of this section follows paper [29], if not otherwise specified. As mentioned previously, the Particle Filter aims to the reconstruction of the entire posterior pdf $p_{\mathbf{X}}(X|Y)$, which in the context of recursive state estimation becomes $p(x(n)|Y^n)$. The underlying algorithm is known as Sequential Importance Sampling (SIS) and is a Monte Carlo method which represents the core of most sequential Monte Carlo filters (SMC); the Particle Filter is such a SMC filter.

The Particle Filter represents the posterior pdf $p(x(n)|Y^n)$ with a set of N_P random samples, or particles, with associated weights; using the notation $\mathcal{X}_i(n)$ for sample i at time n, and $w_i(n)$ for the corresponding weight, analogously to the notation adopted for the UKF, the set of particles and weights at time n is given by:

$$\{\mathcal{X}_i(n),\ w_i(n)\}_{i=1}^{N_p}.$$

Like for the UKF, the weights are normalized, i.e. $\sum_{i=1}^{N_p} w_i(n) = 1$. Using these weighted particles, the posterior pdf is approximated by the following discrete pdf:

$$\tilde{p}_{N_P}(x(n)|Y^n) = \sum_{i=1}^{N_p} w_i(n)\ \delta(x(n) - \mathcal{X}_i(n)) \approx p(x(n)|Y^n). \tag{4.13}$$

At first appearance this approach seems to be really similar to the UKF approach; anyway, there is a fundamental difference between the particles of the PF and the sigma points of the UKF. The particles are randomly chosen, whereas the sigma points are designed carefully, according to a deterministic procedure. Therefore, if the number of particles is the same as the number of sigma points, UKF will yield better performance, since its points are chosen systematically. On the other hand, for the PF it can be proved that for a large number of particles the posterior pdf is matched *exactly*, i.e.:

$$\lim_{N_P \to \infty} \tilde{p}_{N_P}(x(n)|Y^n) = p(x(n)|Y^n).$$

Returning to the sampling problem, unfortunately in most cases it is not possible to sample from the desired pdf $p(\cdot)$ (in our case, the true posterior pdf): for this reason the principle of importance sampling (IS) is introduced. More precisely, a so-called importance or proposal pdf $q(\cdot)$ is chosen, which is easy to sample; the particles $\{\mathcal{X}_i\}_{i=1}^{N_p}$ are then drawn from this pdf. It must satisfy certain properties listed in [29]; the most important constraint concerns the support. As a matter of fact, the support of $q(\cdot)$ must contain the one of $p(\cdot)$ [34]. For the rest, the importance pdf can be chosen almost arbitrarily; in this paper, two particular choices will be considered.

Adopting the importance sampling approach, the weights in the discrete approximation of $p(\cdot)$,

$$\tilde{p}_{N_P}(x) = \sum_{i=1}^{N_p} w_i \, \delta(x - \mathcal{X}_i),$$

are chosen as follows:

$$w_i \propto \frac{p(\mathcal{X}_i)}{q(\mathcal{X}_i)}, \qquad i = 1,\, 2,\, \ldots,\, N_P. \tag{4.14}$$

Here the proportionality symbol was used, since the weights have to be normalized. Note that the pdfs in this equation are evaluated in the points defined by the random samples.

Returning to the case of interest of recursive state estimation, more precisely of filtered state estimation at each time n, the importance sampling becomes sequential (SIS), and the desired and the importance pdf respectively:

$$\begin{aligned} p(\cdot) &\;\to\; p(x(n)|Y^n), \\ q(\cdot) &\;\to\; q(x(n)|x(n-1),\, y(n)). \end{aligned}$$

Lets consider the set of weighted particles, drawn at time $n-1$,

$$\{\mathcal{X}_i(n-1),\, w_i(n-1)\}_{i=1}^{N_p},$$

which approximates the true posterior pdf $p(x(n-1)|Y^{n-1})$ at time $n-1$. Then N_p new samples are drawn from $q(x(n)|x(n-1),\ y(n))$, and the weights are updated according to the following rule (for a detailed treatment refer to [29]; basically, the Bayesian rule is performed):

$$w_i(n) \propto w_i(n-1)\frac{p(y(n)|\mathcal{X}_i(n))\ p(\mathcal{X}_i(n)|\mathcal{X}_i(n-1))}{q(\mathcal{X}_i(n)|\mathcal{X}_i(n-1),\ y(n))}, \qquad i=1,\ 2,\ \ldots,\ N_P.$$

Finally, the new weights are normalized and the posterior pdf at time n is approximated according to equation (4.13). The computation of its first two moments, i.e. estimation $\hat{x}(n|n)$ and estimation error covariance $P(n|n)$, is straightforward:

$$\begin{aligned}\hat{x}(n|n) &= \sum_{i=1}^{N_P} w_i(n)\ \mathcal{X}_i(n),\\ P(n|n) &= \sum_{i=1}^{N_P} w_i(n)\ (\mathcal{X}_i(n)-\hat{x}(n|n))\ (\mathcal{X}_i(n)-\hat{x}(n|n))^T.\end{aligned}$$

These steps are repeated at each iteration. The algorithm is summarized at the end of the section, for a particular choice of importance density and with resampling. Finally, it is worthwile to mention that the measurement update in the case of the PF is performed by the weight update. The time update, i.e. the prediction, is obtained performing the pdf

$$p(\mathcal{X}_i(n)|\mathcal{X}_i(n-1)),$$

which appears in the weight update equation; more precisely, the predicted particles $\{\mathcal{X}_i(n|n-1)\}_{i=1}^{N_P}$ are obtained by applying the particles $\{\mathcal{X}_i(n-1)\}_{i=1}^{N_P}$ to the state update equation:

$$\mathcal{X}_i(n|n-1) = f(\mathcal{X}_i(n-1),\ u(n-1), \mathcal{W}_{\mathcal{X},i}(n-1)) \quad i=1,\ \ldots,\ N_P. \tag{4.15}$$

Here $\mathcal{W}_{\mathcal{X},i}(n-1)$ are the random samples drawn from the process noise pdf, $\mathcal{N}(0,\ Q)$ in our case. Then state prediction and prediction error covariance are computed using the *old* weights:

$$\begin{aligned}\hat{x}(n|n-1) &= \sum_{i=1}^{N_P} w_i(n-1)\ \mathcal{X}_i(n|n-1),\\ P(n|n-1) &= \sum_{i=1}^{N_P} w_i(n-1)\ (\mathcal{X}_i(n|n-1)-\hat{x}(n|n-1))\ (\mathcal{X}_i(n|n-1)-\hat{x}(n|n-1))^T.\end{aligned}$$

Therefore, Particle Filtering relies on the same time update - measurement update structure as Kalman filtering. Furthermore, the PF requires no assumptions on state and noise pdfs; that is, it can be applied to every type of densities and nonlinear model dynamics. It is not restricted to Gaussian or unimodal pdfs, and furthermore it approaches, for $N_P \rightarrow \infty$, the true posterior pdf and therefore the true mean and covariance (it follows from the *Law of large numbers* [34]). However, it is difficult to handle and computationally expensive; the computational cost increases with the number of particles.

4.8.1 Degeneracy problem and resampling

The SIS algorithm as presented above suffers from the so-called degeneracy problem: that is, after a few iterations almost all particles have negligible weight, and typically there is only one particle left with significant weight (close to 1). Intuitively, this is due to the variance of the weights, which can only increase over time [29]; renouncing on a detailed description, which can be found in [30], it can be stated that the variance would be zero if the particles were drawn from the desired posterior pdf, and it increases as the importance density approximation of the desired density worsens. Therefore, an intelligent choice of the importance pdf is fundamental. Beside a good choice of $q(\cdot)$, resampling is introduced to reduce the variance increment.

Obviously, the situation described above is undesired since a large computation effort is spent to propagate many particles with low contribution (i.e., small weights) to the discrete pdf; practically, they are almost useless. The effective number of samples which contribute to the discrete pdf is defined as:

$$N_{eff} = \frac{N_p}{1 + Var[w_i^*(n)]}.$$

The asterisked weights are the true weights defined by equation (4.14); if $q(\cdot) = p(\cdot)$ they are all equal to 1 ($\frac{1}{N_P}$ after the normalization) and therefore the variance would be zero, leading to $N_{eff} = N_P$. However, usually this is not the case and furthermore the equation can not be evaluated exactly; therefore an approximation is adopted [30]:

$$N_{eff} \approx \frac{1}{\sum_{i=1}^{N_P} (w_i(n))^2}.$$

Again, if all the weights have the same value, $\frac{1}{N_P}$, then $N_{eff} = N_p$; in the opposite case, when all the weights except one have values close to zero, then $N_{eff} = 1$, since the only significant particle will have weight 1, as mentioned above. Therefore, the effective sample size lies between 1 (worst case) and N_P (best case). Usually a threshold N_T is defined, and if $N_{eff} \leq N_T$, the particles are resampled.

The resampling is based on the following idea: it is better to eliminate particles with small weights and to replicate particles with large weights. To achieve this the degenerated particle set $\{\mathcal{X}_i(n)\}_{i=1}^{N_P}$ is replaced with the new set[17] $\{\mathcal{X}_k^*(n)\}_{k=1}^{N_P}$, obtained by (re-)sampling N_P times from the discrete distribution (4.13)

$$\tilde{p}(x(n)|Y^n) = \sum_{i=1}^{N_p} w_i(n)\, \delta(x(n) - \mathcal{X}_i(n)),$$

used to approach the true posterior pdf. Note that a particle $\mathcal{X}_i$ from the degenerated set can be picked more than once (and will, if it has a large weight), i.e. the sampling is executed with replacement. The whole procedure is characterized by the probability:

$$P(\mathcal{X}_k^*(n) = \mathcal{X}_i(n)) = \tilde{p}(\mathcal{X}_i(n)|Y^n) = w_i(n).$$

From this expression it is clear that particles with large weights will be sampled more often than particles with small weigths. The resampled set is an indipendent, identically distributed (i.i.d.) sample from the discrete pdf and therefore every particle $\mathcal{X}_k^*(n)$ is assigned the same weight $w_k^* = \frac{1}{N_P}$. Many different resampling algorithms have been developed in the last years: amongst others, the best known are *Multinomial resampling*, *Stratified resampling*, *Systematic resampling* and *Residual resampling*. A detailed comparison of these algorithms is proposed in [31], where Systematic resampling is considered the best approach with respect to resampling quality and computational efficiency. Therefore, here only this method was considered.

Systematic Resampling

Lets consider a finite discrete pdf like (4.13), denoted as:

$$p_{\mathbf{x}}(x) = \sum_{i=1}^{N} w_i\, \delta(\mathbf{x} - \mathcal{X}_i). \tag{4.16}$$

Its cumulative distribution function (CDF) is defined as:

$$F_{\mathbf{x}}(x) = P_{\mathbf{x}}(\mathbf{x} \leq x).$$

That is, $F_{\mathbf{x}}(x)$ is the probability that $\mathbf{x}$ is less or equal then x; in the case of interest of a discrete pdf it becomes:

$$F_{\mathbf{x}}(x) = \sum_{i:\mathcal{X}_i \leq x} p_{\mathbf{x}}(\mathcal{X}_i) = \sum_{i:\mathcal{X}_i \leq x} w_i.$$

---

[17] the index was changed from i to k to improve the readibility of the following considerations

Now, a (pseudo-)random sample can be drawn from $p_{\mathbf{x}}(x)$ according to the following, well known *inverse transform method for finite discrete distributions* [32]:

1. A random sample u is generated from the standard (continuous) uniform distribution $\mathcal{U}[0,1]$ in the interval $[0,\ 1]$.

2. The value $\mathcal{X}^* \in \{\mathcal{X}_i\}_{i=1}^{N}$ is computed, such that

   $$F_{\mathbf{x}}(\mathcal{X}_{i-1}) \leq u < F_{\mathbf{x}}(\mathcal{X}_i),$$

   or, equivalently, such that:

   $$\sum_{s=1}^{i-1} w_s \leq u < \sum_{s=1}^{i} w_s.$$

   That is, from a formal point of view, $\mathcal{X}^*$ is obtained by performing the generalized inverse of $F_{\mathbf{x}}$ in u.

It can be proved that the sample $\mathcal{X}^*$ generated by this algorithm is a random sample drawn from the discrete pdf (4.16); obviously, from the expression of $p_{\mathbf{x}}(x)$ it holds that:

$$P_{\mathbf{x}}(\mathcal{X}^* = \mathcal{X}_i) = p_{\mathbf{x}}(\mathcal{X}_i) = w_i.$$

Systematic sampling uses this approach in order to obtain a random sample from the distribution (4.13); it is repeated N_P times, in order to generate N_P random samples. However, only at the first iteration a uniformly distributed number, lets call it u_1, is generated; for the successive iterations, the (no longer uniformly distributed) numbers are generated as follows:

$$u_k = \frac{(k-1) + u_1}{N_P}, \qquad k = 2,\ 3,\ \ldots,\ N_P. \tag{4.17}$$

This yields an ordered sequence $(u_1,\ u_2, \ldots u_{N_P})$; on the contrary, in the Multinomial resampling algorithm each element of the sequence is uniformly distributed; this is the fundamental difference between the two approaches.

This subsection is concluded with a summary of the Systematic Resampling algorithm, implemented for this project. Given the degenerated set $\{\mathcal{X}_i(n),\ w_i(n)\}_{i=1}^{N_P}$, the resampled set $\{\mathcal{X}_k^*(n),\ w_k^*(n)\}_{k=1}^{N_P}$ is obtained as follows:

1. The ordered sequence $(u_1,\ u_2, \ldots u_{N_P})$ is generated, where u_1 is drawn from the standard uniform distribution and the other elements are generated according to (4.17).

2. For $k = 1, 2, \ldots, N_P$, $\mathcal{X}_k^*(n)$ is chosen as follows:

$$\mathcal{X}_k^*(n) = \mathcal{X}_i(n), \quad \text{with } i \text{ such that} \quad u_k \in \left[\sum_{s=1}^{i-1} w_s(n), \; \sum_{s=1}^{i} w_s(n)\right).$$

3. The weights $w_k^*(n)$ of the resampled particles are all set to $\frac{1}{N_P}$.

Hence, the degenerated posterior pdf approximation

$$\tilde{p}_{N_P}(x(n)|Y^n) = \sum_{i=1}^{N_p} w_i(n)\, \delta(x(n) - \mathcal{X}_i(n)),$$

is replaced by [31]:

$$\tilde{p}^*_{N_P}(x(n)|Y^n) = \sum_{k=1}^{N_p} w_k^*(n)\, \delta(x(n) - \mathcal{X}_k^*(n)) = \sum_{i=1}^{N_p} \frac{n_i}{N_P}\, \delta(x(n) - \mathcal{X}_i(n)).$$

Here, n_i is the number of copies of particle $\mathcal{X}_i$ from the degenerated set contained in the resampled set. Finally, it is worthwhile to mention that this resampling procedure is equivalent to sample n_i from a multinomial distribution described by the probability [33]:

$$P_{\mathbf{x}}(x = \mathcal{X}_i) = w_i.$$

Details on the multinomial distribution can be found in each probability textbook.

Unfortunately, resampling has also an undesired side effect; since the particles with large weights are picked more often, and particles with small weights are discarded, it reduces the number of distinct particles. This phenomenon is known as *loss of diversity* or *sample impoverishment* and there exist several improvements of the Particle Filter algorithm which counteract this effect. For this project the *loss of diversity* problem was neglected; anyway, it is treated in [29].

4.8.2 Choice of the importance density

As mentioned earlier, the choice of the importance density is the crucial point of the Particle Filter design. A myriad of importance densities can be found in literature; a common, intuitive choice is the prior density [29]

$$q(x(n)|x(n-1),\, y(n)) = p(x(n)|x(n-1)),$$

i.e. the pdf of $x(n)$ given $x(n-1)$. Sampling from this pdf is straightforward, according to equation (4.15).
With this choice for the importance pdf the weight update equation

$$w_i(n) \propto w_i(n-1)\frac{p(y(n)|\mathcal{X}_i(n))\ p(\mathcal{X}_i(n)|\mathcal{X}_i(n-1))}{q(\mathcal{X}_i(n)|\mathcal{X}_i(n-1),\ y(n))}, \quad i = 1,\ 2,\ \ldots,\ N_P,$$

simplifies to:

$$w_i(n) \propto w_i(n-1)\ p(y(n)|\mathcal{X}_i(n)), \qquad i = 1,\ 2,\ \ldots,\ N_P.$$

The pdf $p(y(n)|\mathcal{X}_i(n))$ is easy to evaluate in the case of purely additive, Gaussian distributed measurement noise. For our capillary model, the measurement equation is $y(n) = H\ x(n) + w_y(n)$, with $H = [0\ 0\ 1]$, and therefore:[18]

$$p_y(y(n)|\mathcal{X}_i(n)) = p_{w_y}(w_y(n)) = p_{w_y}(y(n) - H\ \mathcal{X}_i(n)).$$

Since $p_{w_y}(\cdot) = \mathcal{N}(0,\ R)$, the pdf results (remember that w_y is a scalar random variable):

$$p_y(y(n)|\mathcal{X}_i(n)) = \frac{1}{\sqrt{2\ \pi\ R}}\ exp\left(-\frac{1}{2}\frac{(y(n) - H\ \mathcal{X}_i(n))^2}{R}\right).$$

Therefore, this choice of importance density is intuitive and simple to implement. However, it might not be a good choice in certain cases.

From a theoretical point of view, the optimal importance density function is the one which reduces the variance of the particle weights and thus maximizes N_{eff} [29]. This optimal density is:

$$q_{opt}(x(n)|x(n-1),\ y(n)) = p(x(n)|x(n-1),\ y(n)).$$

However, usually it is not possible to sample from this pdf. Anyway, there exists an important exception, namely when it is Gaussian and therefore sampling is straightforward. This happens when the measurement equation of the filter model is linear and the process and measurement noises are Gaussian distributed and mutually indipendent, which is exactly the situation of our capillary model. In this case the optimal importance density is given by [29]:

$$q_{opt}(x(n)|x(n-1),\ y(n)) \quad = \quad \mathcal{N}(m,\ P).$$

The involved mean and covariance matrix are given by:

$$P \quad = \quad (Q^{-1} + H^T\ R^{-1}\ H)^{-1}, \tag{4.18}$$

$$m \quad = \quad P\ (Q^{-1}\ f(x(n-1)) + H^T\ R^{-1}\ y(n)). \tag{4.19}$$

---

[18] as it is well known, if $w_y \sim p_{w_y}(w_y)$ and $y = h(w_y)$, then $y \sim |\frac{d}{dy}(h^{-1}(y))|\ p_{w_y}(h^{-1}(y))$; in this case, $h(\cdot)$ is the linear measurement function

Furthermore, the weight update equation becomes

$$w_i(n) \propto w_i(n-1)\; p(y(n)|\mathcal{X}_i(n-1)), \qquad i = 1,\, 2,\, \ldots,\, N_P,$$

where the involved pdf is given by:

$$p(y(n)|x(n-1)) = \mathcal{N}(\bar{m},\, \bar{P}),$$

with:

$$\begin{aligned} \bar{m} &= H\, f(x(n-1)), && (4.20)\\ \bar{P} &= R + H\, Q\, H^T. && (4.21) \end{aligned}$$

This is the choice adopted for this project. The Particle Filter designed with the optimal density function is known as *Gaussian Particle Filter*, since it relies on a partially Gaussian state space representation.

4.8.3 Summary of the Gaussian Particle Filter algorithm

Summarizing, the Gaussian Partice Filter with resampling is structured as follows:

1. If m_0 and P_0 are the initial mean and covariance matrix, like for the Kalman filter, then the initial particle set $\{\mathcal{X}_i(n_0),\, w_i(n_0)\}_{i=1}^{N_P}$ is chosen as follows:

$$\begin{aligned} w_i(n_0) &= \frac{1}{N_P}, \qquad i = 1,\, 2,\, \ldots,\, N_P,\\ \mathcal{X}_i(n_0) &\sim \mathcal{N}(m_0,\, P_0), \qquad i = 1,\, 2,\, \ldots,\, N_P. \end{aligned}$$

2. For $n > n_0$, the new particles $\{\mathcal{X}_i(n)\}_{i=1}^{N_P}$ are sampled as follows:

$$\mathcal{X}_i(n) \sim \mathcal{N}(m,\, P), \qquad i = 1,\, 2,\, \ldots,\, N_P.$$

   m and P are defined by (4.19) and (4.18).

3. Then, the weights are updated, according to:

$$\bar{w}_i(n) = w_i(n-1)\, \mathcal{N}(\bar{m},\, \bar{P}), \qquad i = 1,\, 2,\, \ldots,\, N_P.$$

   $\bar{m}$ and $\bar{P}$ are defined by (4.20) and (4.21).
   Finally, the weights are normalized:

$$w_i(n) = \frac{\bar{w}_i(n)}{\sum_{i=1}^{N_P} \bar{w}_i(n)}, \qquad i = 1,\, 2,\, \ldots,\, N_P.$$

4. The effective sample size is computed from the (normalized) updated weights:

$$N_{eff} = \frac{1}{\sum_{i=1}^{N_P} (w_i(n))^2}.$$

   If $N_{eff} \leq N_T$, where $N_T \leq N_P$ is the pre-defined threshold, the previously explained Systematic resampling algorithm is applied in order to compute the resampled set

$$\{\mathcal{X}_k^*(n),\, w_k^*(n)\}_{k=1}^{N_P},$$

   from the degenerated set:

$$\{\mathcal{X}_i(n),\, w_i(n)\}_{i=1}^{N_P}.$$

5. Finally, the discrete approximation of the true posterior pdf is given by:[19]

$$\tilde{p}_{N_P}^*(x(n)|Y^n) = \sum_{i=1}^{N_p} \frac{n_i}{N_P}\, \delta(x(n) - \mathcal{X}_i(n)).$$

From the posterior pdf approximation of point 5 all required moments can be computed (usually mean and estimation error covariance).

---

[19] if resampling was executed; otherwise, standard expression (4.13) is used

Chapter 5

Estimation quality evaluation: whiteness tests

5.1 The principle of whiteness tests

In the previous chapter the Bayesian estimation problem was discussed, and several state estimation approaches were proposed. However, a fundamental question was not yet adressed: how to evaluate the estimation quality? Some kind of quality index is required in order to tune the filters correctly.

Obviously for estimations on a simulated model the “true“ state evolutions can be compared to the estimated ones; however for real systems the state variables are not accessible and therefore this kind of comparison is not possible. As a matter of fact, in any practical estimation problem the only available information is the measured output variable of the considered system, together with the estimations and predictions computed by the filter.

Based on this set of informations, the standard approach to estimation quality evaluation is the so-called test for white noise on the output prediction error process [23], defined as:

$$e(n) = y(n) - \hat{y}(n|n-1), \qquad n \geq n_0.$$

That is, the difference between the measured output y of the system (the radius measurement in our case) and the output prediction $\hat{y}(n|n-1)$ calculated by the filter. As mentioned in the previous chapter, for linear estimation problems the prediction error computed by the linear MVUE (such as the Kalman filter) is called innovation; if the estimator is tuned correctly, the innovation forms a white noise process. A formal definition of whiteness is given in the next section; basically it means that $\forall n \in \mathbb{Z}$ the autocorrelation function is of the form:

$$r(\tau) = \mathbb{E}[e(n+\tau)\; e(n)] = \begin{cases} M & \text{for } \tau = 0 \\ 0 & \text{for } \tau \neq 0 \end{cases}.$$

M is the statistical power of the innovation, defined as $r(0)$. Obviously $\tau \in \mathbb{Z}$ too, since we are dealing with discrete-time processes.
According to [23], where a formal proof is given, the Kalman filter is tuned correctly, that is its noise covariances are chosen appropriately, if and only if the innovation process $e(\cdot)$ is white noise. In the context of nonlinear estimation, here attempted with the EKF, the UKF and the PF, the prediction error is only approximately white; therefore, it is no longer called innovation. However, the test for white noise remains the standard method for estimation quality evaluation also in this case. Whereas in the linear case the optimal tuning yields perfectly white noise, in the nonlinear case this objective can be achieved only approximately.
There exist several methods in order to evaluate the whiteness of the prediction error $e(\cdot)$. They rely on the estimation of the autocorrelation function, the Power Spectral Density and/or the Power Spectrum. A formal definition of these functions is given in the next section. After that, two estimation approaches are presented and finally two whiteness indices are developed.

5.2 Formal definition of white noise

For the discrete-time stochastic processes considered throughout this project, the following properties were tacitly assumed to be satisfied:

- Real valued with finite variance
- Wide-sense stationary
- Unitary sampling time

The following definitions, based on [37], refer to the scalar output prediction error process $e(\cdot)$, which is supposed to satisfy these properties. Details on the basic properties of stochastic processes, such as stationarity, can be found in each probability textbook. Again, random variables and processes are denoted with normal font.

Lets consider the autocorrelation function $r(\cdot)$ and the autocovariance function $\sigma(\cdot)$ of $e(\cdot)$ (m denotes the mean, as usual):

$$\begin{aligned} r(\tau) &= \mathbb{E}[e(n+\tau)\; e(n)], \qquad n,\ \tau \in \mathbb{Z}, \\ \sigma(\tau) &= \mathbb{E}[(e(n+\tau) - m)\; (e(n) - m)], \quad n,\ \tau \in \mathbb{Z}. \end{aligned}$$

Alternatively, the process $e(\cdot)$ can be described in the frequency domain, more precisely with its Power Spectral Density (PSD) and Power Spectrum (PS), defined as follows.

Definition 1. (**Power Spectral Density**)
The Power Spectral Density $S(e^{j\omega})$ of process $e(\cdot)$ is given by the Discrete-Time Fourier Transform (DTFT) of its autocorrelation function:

$$S(e^{j\omega}) = \sum_{\tau=-\infty}^{\infty} r(\tau)\, e^{-j\omega\tau}, \qquad \omega \in [-\pi,\ \pi].$$

♦

Thus, the Power Spectral Density is the probabilistic counterpart of the deterministic Fourier Transform.

Definition 2. (**Power Spectrum**)
The Power Spectrum $M(\omega)$ of process $e(\cdot)$ is given by the integral of its PSD:

$$M(\omega) = \frac{1}{2\pi}\int_{-\pi}^{\omega} S(e^{j\xi})\, d\xi, \qquad \omega \in [-\pi,\ \pi].$$

♦

The following properties hold for these quantities:

- the autocorrelation $r(\tau)$ is a real valued, even function of τ;
- the PSD $S(e^{j\omega})$ is a real valued, even and non negative function of ω;
- the PS $M(\omega)$ is a real valued, monotonically non-decreasing and limited function of ω; furthermore:

  $$M(-\pi) = 0, \quad \text{and} \quad M(\pi) = r(0) = M.$$

  M is the statistical (time-indipendent) power of process $e(\cdot)$. Therefore, the integral of the PSD over $[-\pi,\ \pi]$, normalized with factor 2π, yields the statistical power of the process.

Based on these quantities the definition of (weak-sense) white noise is given. Let $e(\cdot)$ be a process with zero mean.

Definition 3. (**White noise**)
A zero-mean process $e(\cdot)$ is said to be white noise if its variables are uncorrelated, that is:

$$r(\tau) = \sigma(\tau) = M\,\delta(\tau), \quad \tau \in \mathbb{Z}.$$

Note that for zero-mean processes no distinction between autocorrelation and autocovariance (and between statistical power and variance) is made.

♦

From Definition 1 it follows that the PSD of a white noise process is constant over the whole frequency range; more precisely, $\forall \omega \in [-\pi,\ \pi]$, it is given by the statistical power M, since $r(0) = M$. Finally, the PS is given by a straight line which connects the points $(-\pi,\ 0)$ and $(\pi,\ M)$. The situation is summarized by Figures 5.1, 5.2 and 5.3, which refer to Matlab generated, Gaussian white noise with unitary statistical power. Note that the Figures report the sample functions, that is the functions computed with respect to a finite-length sequence of the process; therefore the theoretical properties of white noise are only approximately matched. Furthermore Figure 5.1 was generated automatically with the Matlab instruction *autocorr()*; it shows only the values for $\tau > 0$, since the function is even.

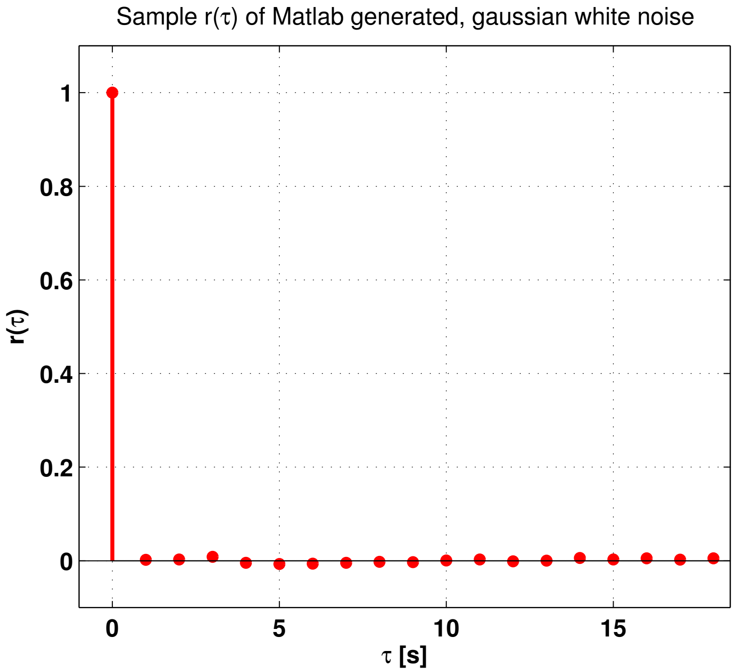

Figure 5.1: Sample autocorrelation function of Matlab generated, Gaussian white noise

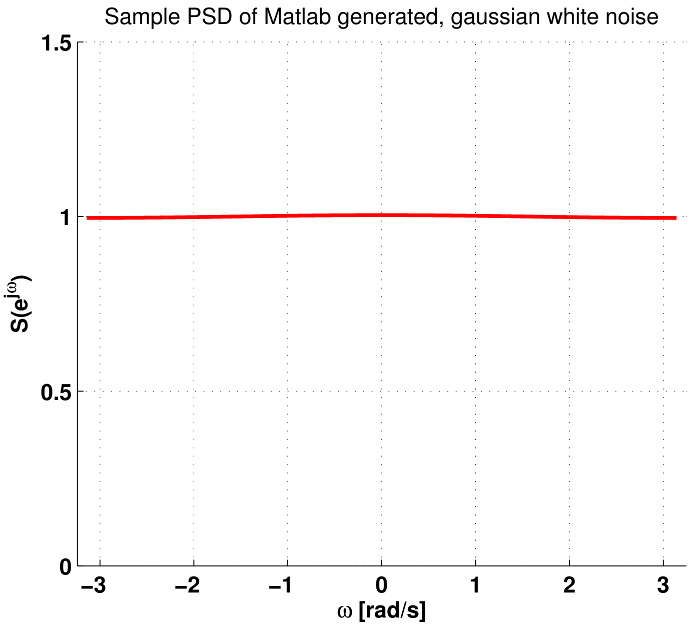

Figure 5.2: Sample PSD of Matlab generated, Gaussian white noise

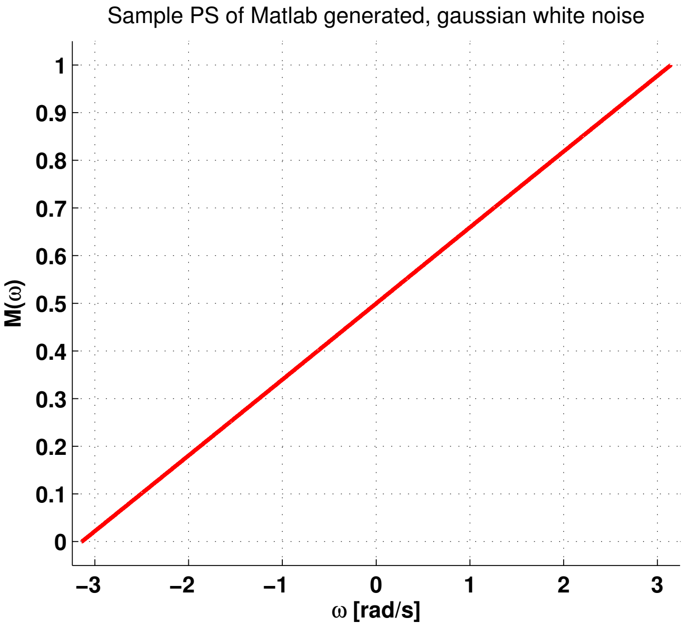

Figure 5.3: Sample PS of Matlab generated, Gaussian white noise

5.3 Autocorrelation, PSD and PS estimation

In order to evaluate the whiteness of a stochastic process, its autocorrelation function, Power Spectral Density or Power Spectrum are compared to the ones of white noise. In this section two approaches to PSD and PS estimation are presented: the first one of non-parametric, the second one of parametric nature. The considerations are loosely based on [28]. For the proper filter tuning only the parametric approach was considered, since it is known to yield better results.
The main problem in the computation of the functions of interest is that for any real problem the only available information on the stochastic process we want to test is given by a finite-length observation. Its probability density function generally is not (entirely) known, and therefore the theoretical autocorrelation function (ACF), PSD and PS can not be evaluated. Thus, the only way to compute these quantities is to *estimate* them from the observed sequence.
Lets consider an observed sequence, or realization, of the output prediction error process $e(\cdot)$, with length N:

$$e(0),\ e(1),\ \dots,\ e(N-1).$$

Non-parametric estimation

Typically, the so-called *sample autocorrelation function* $\hat{r}_N(\cdot)$ is considered for the estimation of the true autocorrelation of $e(\cdot)$. It is defined as follows [28]:

$$\hat{r}_N(\tau) = \begin{cases} \frac{1}{N} \sum_{n=0}^{N-1-\tau} e(n+\tau)\, e(n), & \tau = 0,\ 1, \dots\ N-1 \\ \hat{r}_N(-\tau) & \tau = -(N-1),\ \dots\ -1 \end{cases} . \tag{5.1}$$

However, the sample autocorrelation function is a biased[1] estimator of $r(\cdot)$. Furthermore, since the PSD is defined as the Discrete Time Fourier Transform of the ACF, it is reasonable to define the PSD estimation analogously, except considering the sample ACF:

$$\hat{S}_N(e^{j\omega}) = \sum_{\tau=-(N-1)}^{N-1} \hat{r}_N(\tau)\, e^{-j\omega\tau}, \qquad \omega = \frac{2\pi}{N} k, \tag{5.2}$$

with $k = 0,\ 1, \dots\ N-1$.
Note that the Discrete Fourier Transform (DFT) instead of the Discrete Time

---

[1] an estimator $\hat{\mathbf{X}}$ of random variable $\mathbf{X}$ is said to be unbiased if $\mathbb{E}[\hat{\mathbf{X}}] = \mathbb{E}[\mathbf{X}]$. Unbiasedness is one of the most important properties for an estimator

Fourier Transform (DTFT) was considered in this definition, that is a discrete frequency domain is assumed, unlike for the theoretical definition of the PSD. This is reasonable since the estimation is calculated on a computer, which can not deal with continuous valued time or frequency domains.
Finally, using expression (5.1) in (5.2) yields

$$\hat{S}_N(e^{j\omega}) = \frac{1}{N}\left|\sum_{n=0}^{N-1} e(n)\, e^{-j\omega n}\right|^2 = \frac{1}{N}\left|E_N(e^{i\omega})\right|^2, \tag{5.3}$$

where $E_N(e^{i\omega})$ is the DFT of the observed sequence $e(0), \dots\ e(N-1)$; that is, $\hat{S}_N(e^{j\omega})$ is computed as the square Euclidean norm of the DFT of the sequence.
The PSD estimator defined by equation (5.3) is the so-called *Periodogram*. Intuitively, the larger is N (that is, the longer is the observed sequence), the better is the estimation of the true prediction error PSD $S(e^{j\omega})$. Furthermore, for $N \to \infty$ one expects the Periodogram to match the true function exactly. However, this is not the case; as a matter of fact, the statistical properties of this estimator are quite poor [28], more precisely:

- It is biased:
  $$\mathbb{E}[\hat{S}_N(e^{j\omega})] \neq \mathbb{E}[S(e^{j\omega})].$$
  Furthermore, it is asymptotically unbiased, that is:
  $$\lim_{N\to\infty} \mathbb{E}[\hat{S}_N(e^{j\omega})] = \mathbb{E}[S(e^{j\omega})].$$
  However this property is not of practical significancy.
- It is not consistent:[2]
  $$\lim_{N\to\infty} \hat{S}_N(e^{j\omega}) \neq S(e^{j\omega}).$$

More precisely the Periodogram has the following asymptotic properties:

- Asymptotical variance:
  $$\lim_{N\to\infty} Var[\hat{S}_N(e^{j\omega})] = \begin{cases} 2\, S^2(e^{j\omega}) & \text{for } \omega = 0,\ \pm\pi \\ S^2(e^{j\omega}) & \text{for } 0 < |\omega| < \pi \end{cases}.$$
- Asymptotical autocorrelation equal to zero for every $\omega_1 \neq \omega_2$.

---

[2] for a more formal definition of consistency refer to [28]; consistency is one of the most important properties for an estimator

That is, asymptotically the Periodogram behaves like a white noise in the frequency domain, with a standard deviation of the order of the true PSD! Obviously an estimator with such a poor performance is, from a practical point of view, useless.

There exist several methods to improve the statistical properties of the Periodogram, that is to reduce its variance. A standard approach is the Bartlett method; it splits the N observations in L subsequent intervals of M samples and computes the Periodogram $\hat{S}_{i,M}(e^{j\omega})$ on each interval, $i = 1,\ 2, \ldots,\ L$. Finally, the average of the L Periodograms is computed, that is:

$$\hat{S}_B(e^{j\omega}) = \frac{1}{L}\sum_{i=1}^{L}\hat{S}_{i,M}(e^{j\omega}).$$

$\hat{S}_B(e^{j\omega})$ is the so-called *Bartlett PSD estimator*. Its bias is L times the bias of the Periodogram, but its variance is reduced by the same factor. Therefore, the number of intervals L has to be chosen carefully in order to get a reasonable tradeoff between bias and variance.[3]

By adopting this approach the oscillations in the PSD estimation are reduced. Finally the Power Spectrum is obtained from the PSD estimation by numerical integration; thus, on the PS estimation the oscillations are once again reduced. However, it is far better to rely on parametric estimation methods.

Parametric estimation

The simpliest parametric approach to PSD and PS estimation is to identify an autoregressive model of order k for the output prediction error $e(\cdot)$:[4]

$$e(n) + a_1\, e(n-1) + \ldots + a_k\, e(n-k) = \epsilon(n). \tag{5.4}$$

$\epsilon(\cdot)$ is a white noise process with variance/power σ^2. If

$$a_1 = \ldots = a_k = 0, \tag{5.5}$$

it holds that $e(\cdot) = \epsilon(\cdot)$ is white noise too.

Applying the Z-transform to model (5.4), it results:[5]

$$\begin{aligned} e(n) + a_1\, z^{-1}\, e(n) + \ldots + a_k\, z^{-k}\, e(n) &= \epsilon(n),\\ e(n)\ [1 + a_1\, z^{-1} + \ldots + a_k\, z^{-k}] &= \epsilon(n). \end{aligned}$$

---

[3] furthermore, there exist several more sophisticated windowing approaches, reported in [28]

[4] that is, a parametric model; hence the denomination *parametric estimation*

[5] the following notation is symbolic; as a matter of fact, it mixes up the Z-domain with the time domain

Defining the polynomial

$$A(z) = 1 + a_1\, z^{-1} + \ldots + a_k\, z^{-k},$$

the output prediction error can be expressed as follows:

$$e(n) = \frac{1}{A(z)}\, \epsilon(n).$$

As it is well known from the properties of the Z-transform, setting $z = e^{j\omega}$ equals to switch from the Z-domain to the frequency domain. In our case this switch yields:

$$e(n) = \frac{1}{A(e^{j\omega})}\, \epsilon(n).$$

Finally, according to the Wiener-Khinchin theorem [37], the PSD of $e(\cdot)$ can be obtained from the PSD of $\epsilon(\cdot)$, given by $S_\epsilon(e^{j\omega}) = \sigma^2$, as follows:

$$\hat{S}(e^{j\omega}) = \frac{1}{A(e^{j\omega})\, A^*(e^{j\omega})}\, \sigma^2 = \frac{\sigma^2}{|A(e^{j\omega})|^2}. \tag{5.6}$$

The asterisk denotes the complex conjugate of $A(e^{j\omega})$; the last equality holds only for the scalar case. If whiteness condition (5.5) holds, then $\hat{S}(e^{j\omega}) = \sigma^2$ and therefore it is perfectly constant, as it should be. Note that the PSD computed this way is again only an estimation, since the parameters $a_1, \ldots, a_k$ are *estimated* by the identification procedure.

The identification of the autoregressive model (5.4) for $e(\cdot)$ can be obtained with the Matlab instruction *ar()*; furthermore the instruction *arfit()* computes the order k of the model for which the best data fit is obtained.

The parametric PSD estimation (5.6) is known to yield better statistical performance as the non-parametric estimation [28]; therefore, in this project the parametric approach was adopted. Again, the PS estimation is obtained from (5.6) by numerical integration. Finally, note that also for parametric estimation a discrete frequency domain is assumed.

5.4 Whiteness indices

Once the functions of interest were estimated for the prediction error $e(\cdot)$, they are compared to the ideal ones, that is the white noise functions. For instance, Figures 5.4, 5.5 and 5.6 report such a graphical comparison for the normalized ACF, PS and PSD (the estimations were computed with the parametric approach). The normalization factor is the statistical power M of $e(\cdot)$. From the Figures it is obvious that $e(\cdot)$ is not white noise; however, a quantification of its whiteness/non-whiteness is required.

A straightforward approach to white noise testing is the correlogram, that is the graphical representation of the *normalized* sample ACF, as in Figure 5.4. If all the values, except the first one obviously, are situated in the confidence interval defined by the blue lines, the signal is white. The confidence interval is defined by a statistical hypothesis test on the normalized sample ACF. The discussion of these tests is beyond the scope of this project; for details, see [28]. However, with this approach it can be stated whether the considered signal is white noise or not, but it yields no quantification of whiteness in case of negative answer.

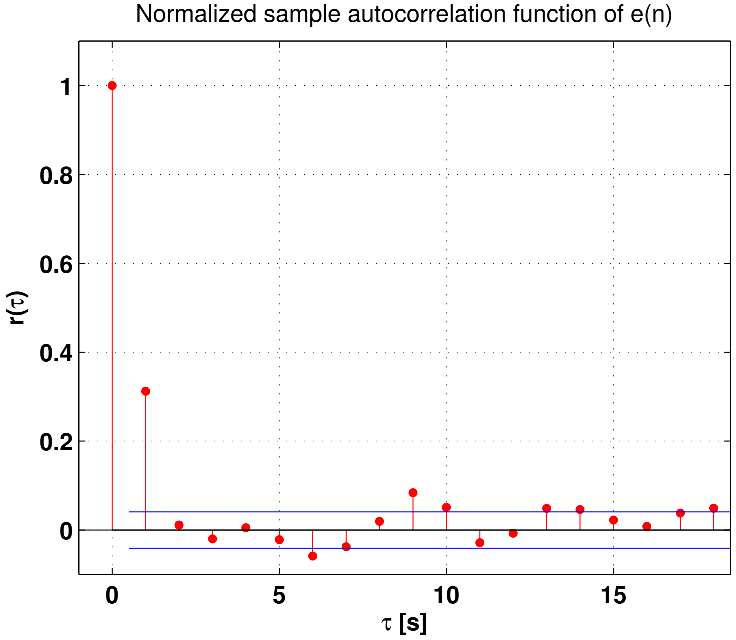

Figure 5.4: Example: Normalized sample autocorrelation function of $e(\cdot)$

Such a quantification can be obtained from Figure 5.6. As a matter of fact, the whiter the prediction error is, the smaller is the area delimited by the ideal PS and the real PS. Therefore, evaluating this area, for $\omega \in [0,\ \pi]$, the whiteness can be quantified. Thus, for this project the following whiteness indices were adopted:

- ***Area index***
  It is defined as the area delimited by the ideal and the real PS, in the interval $[0,\ \pi]$ (this area equals the area computed in the interval $[-\pi,\ 0]$, due to the symmetry of the PSD).

- ***Maximum deviation index***
  It is defined as the maximum deviation of the real PS from the ideal

PS, again on the interval $[0,\ \pi]$.

The closer this indices are to zero, the whiter is the prediction error. Note that they are computed on the normalized PS. Especially the area index was heavily used for filter tuning, as shown in the next chapters.

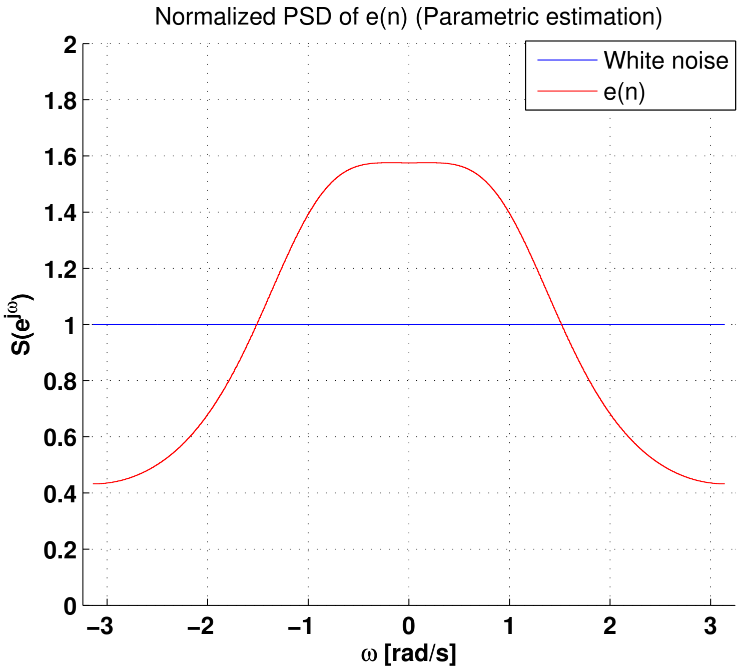

Figure 5.5: Ex.: Normalized parametric PSD of $e(\cdot)$ vs. white noise PSD

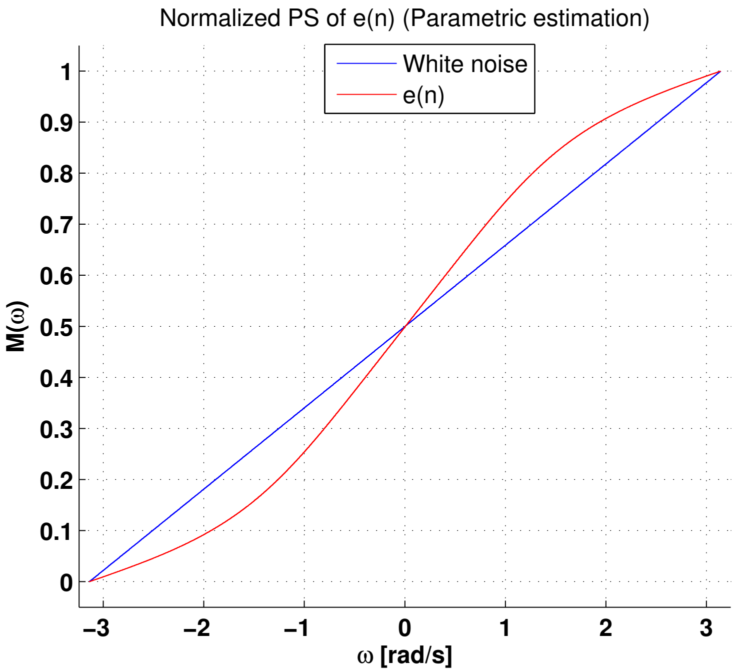

Figure 5.6: Ex.: Normalized parametric PS of $e(\cdot)$ vs. white noise PS

Chapter 6

The infinite-horizon linear-quadratic regulator (LQR)

The classical approach to state feedback control consists in the pole or eigenvalue allocation of the controlled (linear) system. Typically, they are allocated in order to match pre-defined specifications on the system output with respect to the tracking of canonical input signals. Typical examples are specifications on the rise and the settling time of the output signal with respect to a Heaviside step input signal. However, with this approach it is difficult to make considerations on the energy spent by the controller. The typical situation is reported in Figure 6.1; as usual, u denotes the input variable, y the measured variable affected by measurement noise w_y (assumed to be additive), x is the not entirely accessible state vector, estimated by a state estimator (for instance, one of the filters proposed previously). Here the feedback is assumed to be linear. Finally, an external disturbance d could act on the controlled plant.

The infinite-horizon linear-quadratic regulator,[1] briefly LQR, represents a model-based approach to state feedback control. This technique permits to achieve the given specifications, that is, usually, tracking of a reference signal with a desired precision and velocity, by computing the so-called optimal control signal. More precisely, the LQR law computes, amongst all admissible input or control signals, the most energy-saving one. Consider a linear, time-invariant, discrete time system (such as the linearized capillary model):

$$\begin{aligned} x(n+1) &= F\,x(n) + G\,u(n), \\ y(n) &= H\,x(n). \end{aligned} \tag{6.1}$$

---

[1] alternatively, also the finite-horizon LQR regulator can be considered

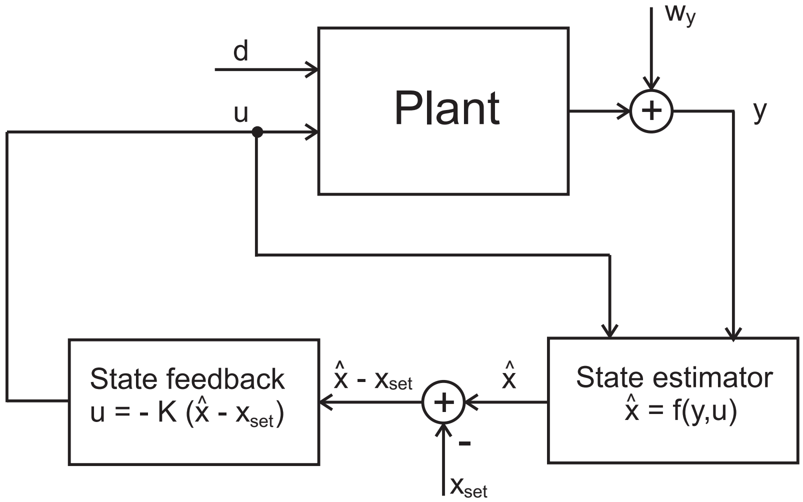

Figure 6.1: Linear state feedback control with state estimator

Let $x(n_0) = x_0$ be the initial condition.
Then, the optimal control signal is computed by minimizing the following, quadratic cost index:

$$J(u,x) = \sum_{n=n_0}^{\infty} [x(n)^T \, Q_{LQR} \, x(n) + u(n)^T \, R_{LQR} \, u(n)]. \tag{6.2}$$

Here, the matrices $Q_{LQR} \in \mathbb{R}^{n \times n}$ and $R_{LQR} \in \mathbb{R}^{m \times m}$ (n is the state space dimension, m the input dimension) are the penalty matrices for the state vector respectively the input vector (which is a scalar in our case). These matrices are the core parameters of the LQR law and will be discussed in the following.
If certain hypothesis are satisfied, the optimal control signal u_{opt} is obtained from a constant, linear feedback of the system state vector. In the next section a brief overview over the LQR theory is given.

6.1 LQR theory review

Based on [38], the basic properties of the infinite-horizon LQR are summarized.
First, it is necessary to understand when does a sequence of inputs $u(n_0)$, $u(n_0 + 1)$, ... exists which yields a finite cost index (6.2). This is a relevant question, since the index is given by the sum of an infinite number of elements. It can be proved that such sequences exist if and only if the controlled system (6.1) is *stabilizable*, that is all the uncontrollable eigenvalues of system matrix F are stable (i.e., their absolute value is < 1). If this

fundamental property is not satisfied, it is not possible to design a LQR controller. As pointed out at the end of Chapter 3, the capillary system is not stabilizable and therefore it has to be modified, as will be shown further on.

If the stabilizability condition is satisfied, then the cost index is minimized by the feedback law

$$u_{opt}(n) = -K_{opt}\, x(n), \qquad \forall n \geq n_0,$$

where the feedback matrix is given by:

$$K_{opt} = (R_{LQR} + G^T\, M\, G)^{-1}\, G^T\, M\, F.$$

M is a positive-semidefinite solution of the so-called *Discrete-time Algebraic Riccati Equation*, briefly DARE, defined as:[2]

$$M = Q_{LQR} + F^T\, M\, F - F^T\, M\, G\, [R_{LQR} + G^T\, M\, G]^{-1}\, G^T\, M\, F.$$

Generally the DARE has more than one positive-semidefinite solution;[3] however, for typical LQR problems it is unique, as pointed out below.

Before addressing this problem, it is worth mentioning that for any feedback law of practical use the system matrix of the feedbacked system,

$$F - G\, K_{opt},$$

has to be stable. As a matter of fact, the optimal feedback law would be worthless if it was not stabilizing. However, it yields a stable system if the conditions summarized by the following theorem are satisfied:[4]

Theorem 3. *Consider a stabilizable, linear system (6.1). If the pair* $(F,\ D)$ *is observable, where* D *is a matrix square root of* Q_{LQR} *such that* $Q_{LQR} = D^T\, D$*, then:*

- *The DARE has only one positive-semidefinite solution* M*, and it is not only positive-semidefinite, but also positive-definite.*
- *For this positive-definite solution the feedback control law described by* K_{opt} *is not only optimal, but also stabilizing.*

◆

---

[2] in Matlab, the solution of the DARE as well as K_{opt} are computed with the instruction *dare(F, G,* Q_{LQR}*,* R_{LQR}*)*

[3] the discussion of which one is the solution to use for the computation of K_{opt} is beyond the scope of this work

[4] if pair $(F,\ D)$ is not observable, but only detectable, which is a weaker property, then the theorem still holds, but the only positive-semidefinite solution of the DARE is not positive-definite

Therefore, if the considered system is stabilizable and Q_{LQR} is chosen in order to satisfy the observability, or at least the detectability condition, then the optimal control law yields a stable system and therefore the LQR can be applied.

Finally, it is worth mentioning that the goal of the standard, finite-horizon LQR is to regulate the state variable/vector to zero, as can be seen from the cost index (6.2); as a matter of fact, only in this case the index can assume a finite value. However, typically the desired state value is not 0, but a non-zero set point x_{set}, which usually is a non-zero equilibrium point of the system, such that:[5]

$$x_{set} = F\ x_{set}.$$

This can be easily achieved by applying a rigid translation to the state space of system (6.1). Lets define the translated state vector:

$$\bar{x} = x - x_{set}.$$

Then the original model (6.1) can be rewritten in the following form (since $x_{set} = F\ x_{set}$):

$$\begin{aligned} [x(n+1) - x_{set}] &= F\ [x(n) - x_{set}] + G\ u(n), \\ \bar{y}(n) &= H\ [x(n) - x_{set}]. \end{aligned}$$

That is the original system, but with translated state vector $\bar{x} = x - x_{set}$. Applying the LQR to this system $\bar{x}$ is regulated to zero, and therefore x is regulated to x_{set}, as desired. The modified feedback law is

$$u_{opt}(n) = -K_{opt}\ [x(n) - x_{set}],$$

where, pay attention, K_{opt} is computed with respect to the original system matrices.
The same approach was adopted for the non-linear capillary model.

6.2 Choice of the penalty matrices

The penalty matrices Q_{LQR} and R_{LQR} are chosen in order to obtain a tradeoff between the regulation precision/promptness of the controller and the control energy expense. The two objectives of promptness and control energy saving are always in contrast: the faster the response is, the larger is the applied control signal, and viceversa.

---

[5] for simplicity, it is supposed to be an equilibrium point with respect to $u = 0$

Considering for the moment both the penalties to be scalar, it can be stated that:[6]

- If $r_{LQR} \gg q_{LQR}$, in the cost index the input signal is penalized more than the state variable. That is, for the minimization of the cost index it is more important to keep the input signal small, whereas deviations of the state variable from the set point are less significant. Therefore the LQR law avoids large control signals, saving on the "control energy". Thus, the control promptness is reduced.
- If $q_{LQR} \gg r_{LQR}$, we are in the opposite situation; that is, the controller is very prompt, spending much energy.

At this point it is worth recalling that for capillary control the control variable is given by the pull rate v (and the (reduced) state vector by $[r \;\; h]^T$). Although a higher pull rate is reflected in a small increase on the energy expense, it is absolutely negligible in the context of crystal pulling. Therefore, the concept of energy is not very appropriate in this case. Nevertheless, also for capillary control it is important to reduce the control signal, that is the pull rate variation; as a matter of fact, whilst the mean value of the pull rate is not that critical for crystal structure and quality, huge instantaneous variations can have a fatal impact. Therefore the principle of two contrasting objectives - pull rate variation reduction vs. prompt radius and height adjustments - still holds.

Now, lets consider the case of matrix penalties; more precisely, the case of interest, where only Q_{LQR} is a matrix. The following considerations are based loosely on [38]. In most cases the state penalty is chosen as a diagonal matrix:

$$Q_{LQR} = \alpha \begin{bmatrix} q_r & 0 \\ 0 & q_h \end{bmatrix}.$$

Here q_r is the penalty on r and q_h the penalty on h. The parameter α is used to scale the state penalty with respect to the control signal penalty R_{LQR}. With this choice of Q_{LQR} the cost index becomes:

$$\begin{aligned}
J(u,x) &= \sum_{n=n_0}^{\infty} \Big((x_r(n) - x_{r,set})^T \, Q_{LQR} \, (x_r(n) - x_{r,set}) + u(n)^T \, R_{LQR} \, u(n) \Big) \\
&= \sum_{n=n_0}^{\infty} \left([r(n) - r_{set} \;\; h(n) - h_{set}] \, \alpha \begin{bmatrix} q_r & 0 \\ 0 & q_h \end{bmatrix} \begin{bmatrix} r(n) - r_{set} \\ h(n) - h_{set} \end{bmatrix} + \Delta v(n)^T \, R_{LQR} \, \Delta v(n) \right) \\
J(u,x) &= \sum_{n=n_0}^{\infty} \Big(\alpha \, (q_r \, (r(n) - r_{set})^2 + q_h \, (h(n) - h_{set})^2) + R_{LQR} \, (\Delta v(n))^2 \Big).
\end{aligned}$$

---

[6] note that the concepts of small and large refer to the absolute value of the input and state variables; the sign does not matter, since the square of the variables is considered in the cost index

From this expression it is clear that q_r acts only on r and q_h only on h, whereas α affects both state variables.
Now the tradeoff between "control energy" and control promptness, explained above for the scalar case, is obtained varying the ratio:

$$\frac{\alpha}{R_{LQR}}.$$

The higher the value of the ratio, the more the LQR focuses on the state evolution, that is, the control promptness and precision improves, tolerating large control signals. If the ratio is kept low, the LQR reduces the control signal, tolerating a "bad" state evolution.
Fixed this ratio, the second ratio,

$$\frac{q_r}{q_h},$$

is considered. Usually a high ratio means that the LQR focuses on the evolution of the first state variable, neglecting the second, and viceversa. However, in the particular case of the capillary model this is not the case; as will be shown in the next chapter, the radius regulation improves (or worsens) together with the meniscus height regulation. It is not possible to improve the radius regulation and at the same time to worsen the height regulation. This unexpected phenomenon is explained by the particular nonlinear relation between r and h, described, for instance, by the Johansen-Boucher formula:[7]

$$h = a\sqrt{\frac{1-sin(\theta)}{1+0.6915\left(\frac{a}{r}\right)^{1.1}}}.$$

In order to achieve constant radius growth it must hold that $\theta = \theta_C$; as a matter of fact, the set points of r and h are chosen in order to satisfy this condition. From the formula it is clear that for an increasing radius the condition can be maintained only if the meniscus height increases too, and viceversa. Therefore, if $|r - r_{set}|$ increases/decreases, the LQR has to increase/decrease the difference $|h - h_{set}|$ too, in order to maintain $\theta = \theta_C$. Thus, if one of the two variables diverges from its set point because it is assigned a low penalty, the other one must diverge too, indipendently from its penalty; this means that it is not possible to regulate r and h indipendently.

This discussion is concluded recalling Bryson's rule for the initial choice of the penalties. With the diagonal structure introduced above, this rule

---

[7] however it can be observed also with the Tsivinskii and the Boucher formula

suggests to use the following penalties:

$$Q_{LQR} = \begin{bmatrix} \frac{1}{(\Delta r_{max})^2} & 0 \\ 0 & \frac{1}{(\Delta h_{max})^2} \end{bmatrix}, \qquad R_{LQR} = \frac{1}{(\Delta v_{max})^2}.$$

Δr_{max}, Δh_{max} and Δv_{max} are the maximum acceptable values of radius, height and pull rate variations (the variations are intended with respect to the set point, for the state variables).
Usually Bryson's rule is used as the starting point for the penalty choice; then the penalties are retuned according to the considerations reported above, in order to obtain the desired results.

Chapter 7

Simulation results

Based on the theory exposed in the previous chapters, the capillary model, the filtering and the control algorithms were implemented. In this chapter the simulation results are shown and discussed: they are based on *simulated* data, that is on artificial data generated with the capillary model. For results on experimental data from St. Peters refer to the next chapter.
As mentioned in the introduction, the algorithms were implemented with *Matlab*,[1] a commercial software developed by *Mathworks*. Matlab is a numerical computing environment designed for complex matrix manipulations and statistical analysis. From Matlab interfacing with programs written in other languages such as *C*, *C++* and *Java* is straightforward. Furthermore Mathworks provides a tool which automatically converts Matlab code into C code.
Matlab is widely used in academic and research institutions, as well as industrial enterprises, especially in the engineering branch, for instance for control theory and digital signal processing. It is provided with an additional package, *Simulink*, a (graphical) tool for modeling, simulating and analyzing dynamic systems. For this project some of the tests on the closed loop system[2] were performed with Simulink; however, the functions called by Simulink are written in Matlab.
In this chapter first some notes on the code implementation are reported, together with the Simulink scheme used for the closed loop simulations, followed by a detailed description of the generated data. After that a detailed comparison between the different filter approaches is reported, together with a step-by-step description of an exemplary tuning procedure. Furthermore a basic robustness analysis was performed. The filter considerations refer to fixed LQR penalties; they are varied in Section 7.3, where some considerations on LQR tuning are exposed.

---

[1] briefly for *Mat*rix *lab*oratory; version *7.8.0.347 (R2009a)* was used

[2] that is the capillary model feed backed with the LQR

7.1 Notes on the implementation in Matlab/Simulink

This section reports a brief summary of the implemented algorithms.

Data generation model and filter model

The data generation model is an intermediate version between the *original* capillary model of [1], given by equations (3.1), and the improved model discussed in Section 3.3. More precisely, it is the original model, but using v_g instead of v in the first state update equation and the Johansen-Boucher formula for θ computation, as proposed in Section 3.3. However, the third state variable is still T_m, that is the random walk is generated with respect to the temperature and not for v_g as in the improved model. This approach was adopted in order to rely on (approximately) the same data as in [1].

On the other hand the filtering algorithms rely on the *improved* model, in order to verify whether the usage of v_g for state estimation is admissible or not. Therefore for filter purposes v_g is considered as the third state variable. Furthermore the filter model includes process noise also on the radius and height update equation: the complete system is described by equations (4.1). Finally also the reduced state space model for LQR purposes is based on the improved model.

Returning to the data generation model, and postponing the discussion of the variance choices for measurement noise and temperature noise to the next section (note that no additional process noise was implemented for the radius and the meniscus height update equation), it is worthwile to describe how the random walk for T_m was implemented. It is generated according to the technique used in [1] and discussed in [2], the so-called *Softclipping*. Basically it bounds the temperature to an interval $[T_{min},\ T_{max}]$; if T_m approaches one of the limits it is moved away according to an exponential function, described by the following law:

- If $T_m < T_{min} + d$, set
$$T_m = T_{min} + d\ \exp\left(\frac{T_m - T_{min}}{d} - 1\right).$$
- If $T_m > T_{max} - d$, set
$$T_m = T_{max} - d\ \exp\left(\frac{T_{max} - T_m}{d} - 1\right).$$

It is reasonable to assume that the temperature is not free to diverge since on the actual puller there is also the second control loop which adjusts the heater power in order to obtain the desired growth rate. Therefore indirectly also the temperature is under control.

For the following simulations, denoting the initial melt temperature with $T_{m,0}$, the function was parameterized as follows:

$$T_{min} = T_{m,0} - 5\ K, \qquad T_{max} = T_{m,0} + 5\ K, \qquad d = 1\ K.$$

The bounded random walk obtained with this function is, according to [2], a quite good representation of the effect of the growth rate control loop.

Filter and LQR implementation

The EKF was implemented according to the algorithm reported in Section 4.5. As stated previously it relies on the same linearization procedure as the LQR algorithm. Furthermore, for the UKF the algorithm of Section 4.6.2 was used, with the sigma point set described in Section 4.6.3. The UKF function is based on the Matlab function written by Yi Cao which can be downloaded from the Mathworks web page [51]. Only some small adjustments where performed, amongst others it was extended to the augmented state space formulation. As a matter of fact, the UKF was implemented with augmented state space.

Finally the Particle Filter was implemented in its Gaussian version, that is the algorithm of Section 4.8.3. Resampling is performed at every iteration, in fact the threshold N_T was set to the total number of particles N_P. The Particle Filter was not tested in the closed loop system (that is "online", where online refers to the simulation obviously) because it is quite expensive from a computational point of view; this is okay since the PF was not implemented for usage in the control system but only for some verifications on the probability density functions of the state vector, in order to verify the EKF and UKF results. Of course for filtering purposes it is indifferent whether state estimation is performed online or offline (on the data generated previously).

Furthermore for estimation quality evaluation only the parametric white noise test was considered, as mentioned in Chapter 5, where also the key instructions were reported.

Figure 7.1 reports the Simulink scheme used for data generation and closed loop simulations. Note that each block of the Simulink scheme calls one or more Matlab functions. The set points are internally fed to the block "LQR". Furthermore the Upsample and Downsample blocks are used for sample time conversion between the model and the feedback branch; however, for the following simulation results they were not used, since the whole loop is characterized by the same sample time, namely 1 second. Finally the scope blocks are used to display the trajectories.

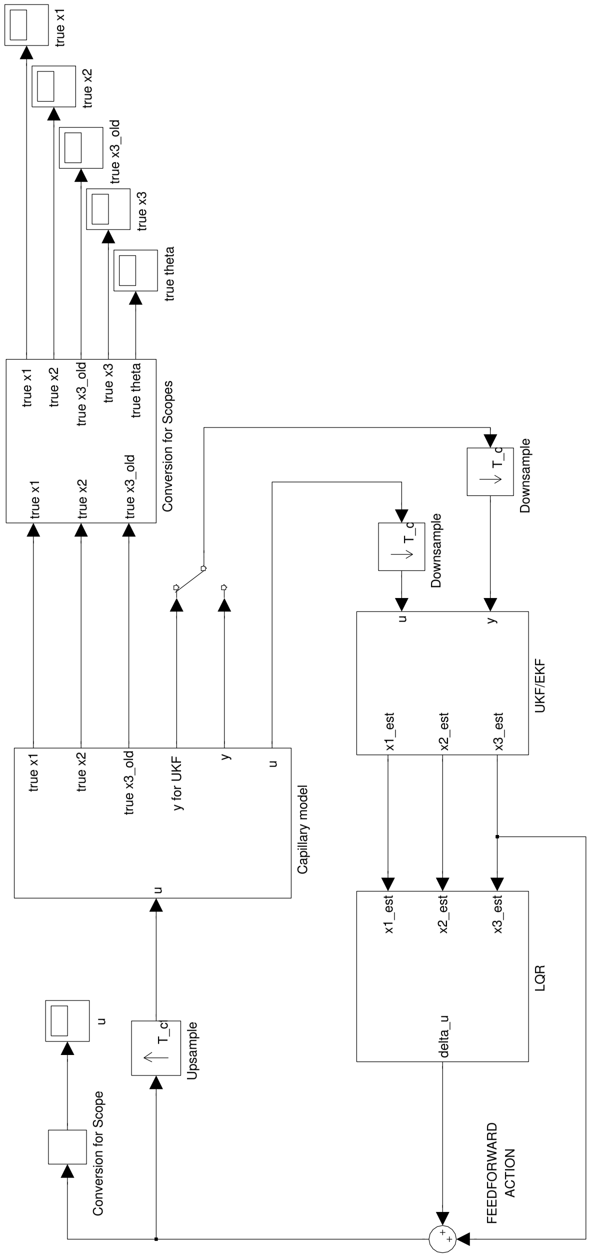

Figure 7.1: Simulink scheme used for closed loop simulations

7.2 Filtering simulations

The filtering tests were performed for fixed LQR penalties, chosen according to Bryson's rule (the exact expressions are not important). Furthermore, the set point

$$\begin{bmatrix} r_{set} \\ h_{set} \end{bmatrix}$$

in this section is assumed to coincide with the initial model conditions. Finally the pull rate variation is limited to $\pm 0.3 \, \frac{mm}{min}$.

All the filtering comparisons were performed on *exactly* the same data; as a matter of fact, the state trajectories were generated only once, using the improved UKF approach for state estimation in the closed loop Simulink scheme.[3] Then the different filter algorithms were applied offline to this data. For statistical estimation performance evaluation this is absolutely legitimate.

The generated reference trajectories are reported in Figures 7.2 - 7.6. Note that the *true* and not the estimated trajectories are shown; furthermore the measured radius is not reported, since it is almost unreadable due to the presence of measurement noise. In the Figures the units typically adopted for crystal pulling are used (mm for radius and height, $^\circ C$ for the melt temperature, $\frac{mm}{min}$ for the growth rate and deg for the meniscus slope angle), which differ from the SI units.

---

[3] the filter parameters for data generation were chosen in order to obtain an acceptable estimation quality; the exact values are not of importance

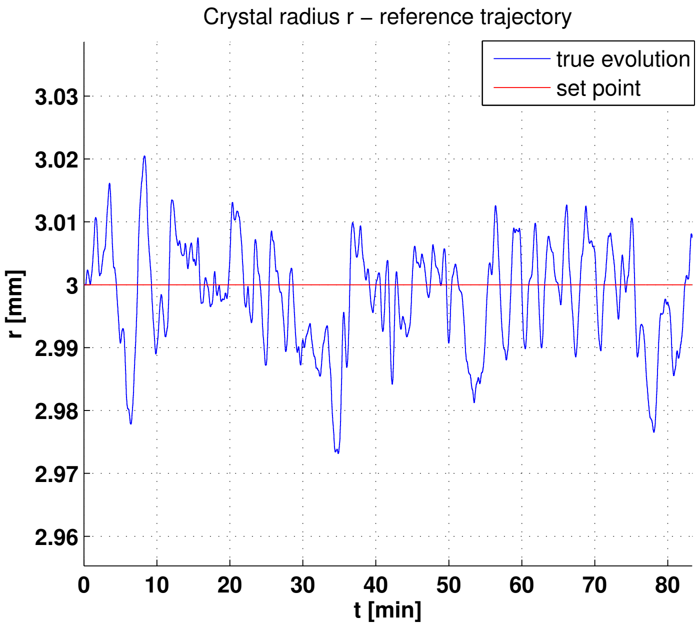

Figure 7.2: Crystal radius r - reference trajectory

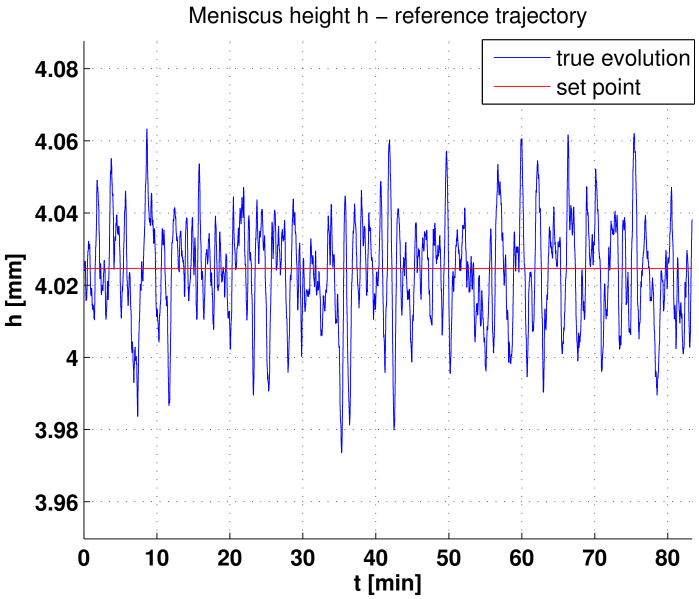

Figure 7.3: Meniscus height h - reference trajectory

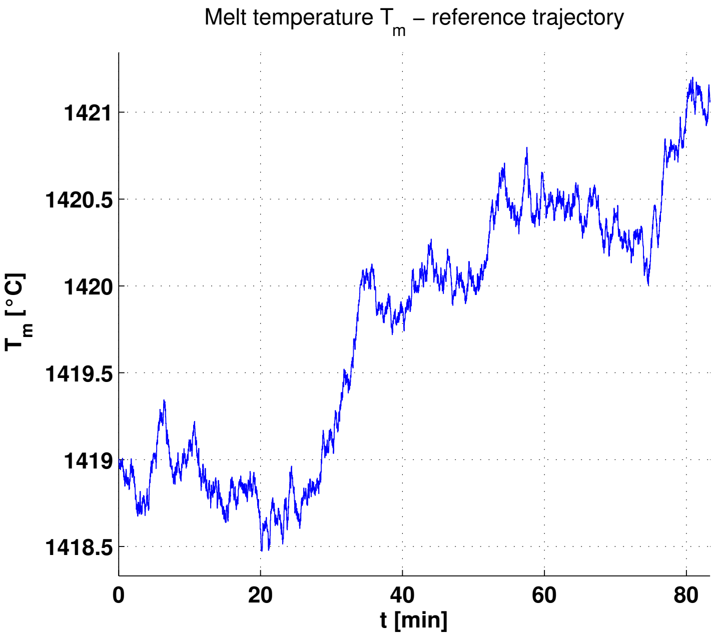

Figure 7.4: Melt temperature T_m - reference trajectory

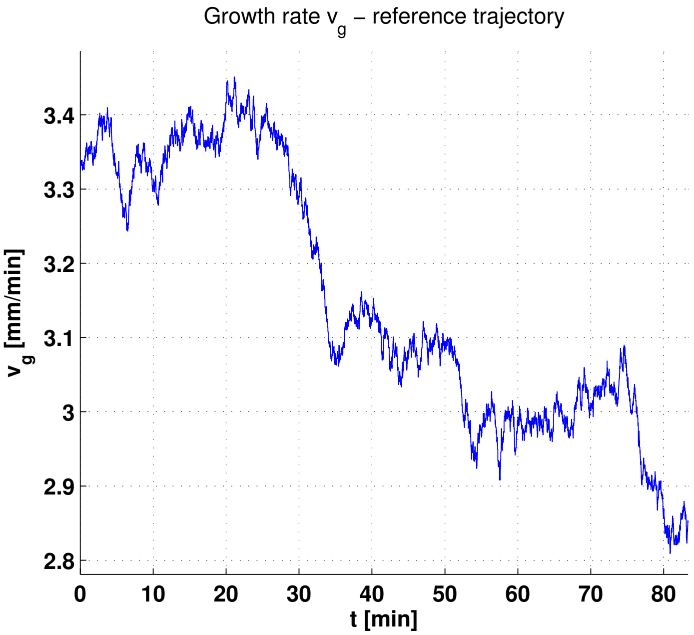

Figure 7.5: Growth rate v_g - reference trajectory

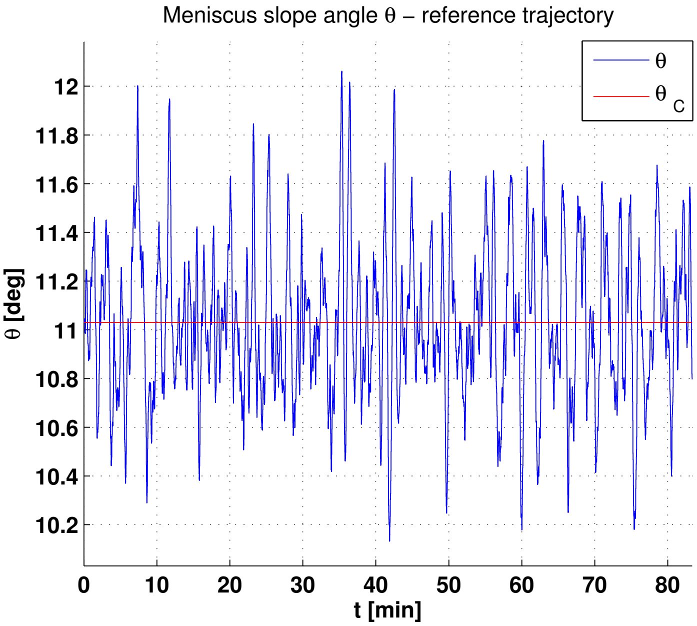

Figure 7.6: Meniscus slope angle θ - reference trajectory

7.2.1 Filter tuning procedure

In the following the typical filter tuning procedure is illustrated. The example refers to the improved UKF, however all filters were tuned according to this procedure which is based on the area index for white noise and of general validity.

Initially, the sigma point parameters are fixed to the standard values:

$$\alpha = 10^{-3}, \qquad k = 0, \qquad \beta = 2.$$

Furthermore the noise variances of the filter,

$$Q = \begin{bmatrix} q_r & 0 & 0 \\ 0 & q_h & 0 \\ 0 & 0 & q_{v_g} \end{bmatrix}, \qquad R,$$

are initialized appropriately. The diagonal entries of Q are the process noise variances for r, h and v_g; R is the measurement noise variance. Note that q_r and q_h are set to a very small value in order to obtain a good matching with the data generation model, for which these variances were set to 0; however

0 is not admissible for the UKF model since Q has to be a positive-definite matrix. Furthermore q_{v_g} was computed, with the growth rate formula (3.2), from the melt temperature variance assumed for data generation.

The filter is initialized with the true initial conditions of the capillary model, that is $\hat{x}(0|0) = x(0)$; this approach was adopted for all the simulations discussed in this chapter. Furthermore the initial covariance $P(0|0)$ was chosen appropriately. However for the white noise test the initial part of the prediction error $e(n)$ is not used, since these tests should not depend on the initial conditions. More precisely, the first 1200 samples (that is, the first 20 minutes) were not used. Therefore the initial conditions are not critical for the filter tuning.

Now for the noise variance tuning it is convenient to factorize Q as follows:

$$Q = q \begin{bmatrix} \bar{q}_r & 0 & 0 \\ 0 & \bar{q}_h & 0 \\ 0 & 0 & \bar{q}_{v_g} \end{bmatrix}.$$

Based on this representation, the first step consists in the optimization of the ratio:

$$\frac{q}{R} = 10^{m_1}.$$

That is, the measurement noise variance is adjusted with respect to q. The optimal choice is the one which minimizes the area index. In the following I_A denotes the value of the area index, multiplied with factor 1000 (this convention was adopted because it is more comfortable to deal with numbers which are greater than unity). Lets assign the following values to R:

$$\left\{10^{-6},\ 10^{-7},\ 10^{-8},\ 10^{-9},\ 10^{-10},\ 10^{-11},\ 10^{-12},\ 10^{-13}\right\}.$$

Now the UKF is applied to the data for each of these configurations; the corresponding values of I_A are reported in Figure 7.7. Note that m_1 is computed as:

$$m_1 = \log_{10} q - \log_{10} R.$$

I_A is minimized for $m_1 = -4$, that is for $R = 10^{-10}$; therefore the best choice for R lies next to this value, as expected, since it is the value used for data generation. In this case $I_A \approx 9$. Further investigation, refining the partition on the interval $[10^{-9},\ 10^{-11}]$, yields that the best choice is:

$$R = 2.5 \cdot 10^{-10}, \quad \text{for which} \quad I_A < 1.$$

Now R is fixed to this optimal value, and the ratio

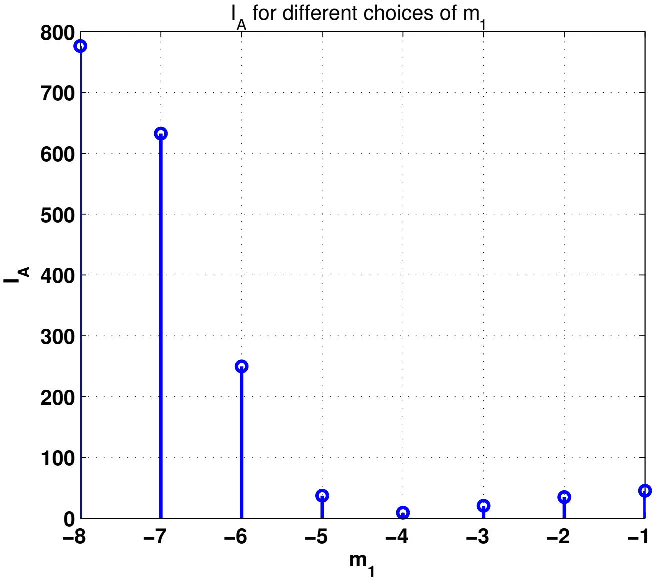

Figure 7.7: I_A for different choices of m_1

$$\frac{q_h}{q_{v_g}} = 10^{m_2}, \quad \text{with} \quad m_2 = \log_{10} q_h - \log_{10} q_{v_g},$$

is considered. Figure 7.8 was obtained with the following values for q_{v_g}:

$$\left\{10^{-11},\ 10^{-12},\ 10^{-13},\ 10^{-14},\ 10^{-15},\ 10^{-16},\ 10^{-17}\right\}.$$

The minimum is obtained for $m = -16$, that is for $q_{v_g} = 10^{-14}$; this is confirmed by a refined minimum search around this value. Therefore the initial choice of q_{v_g} was appropriate, and the minimum index is still the same.

The next step considers the ratio:

$$\frac{q_r}{q_h} = 10^{m_3}, \quad \text{with} \quad m_3 = \log_{10} q_r - \log_{10} q_h.$$

Figure 7.9 was obtained assigning the following values to q_h:

$$\left\{10^{-12},\ 10^{-13},\ \ldots,\ 10^{-30}\right\}.$$

From the Figure it is clear that the minimum is obtained for $m_3 = -17$; for this value $I_A < 0.1$.

Finally q_r is tuned. Omitting the graphical representation and the intermediate steps, it can be affirmed that for $q_r \leq 10^{-19}$ the index value does not change, whereas it increases slightly for higher values.

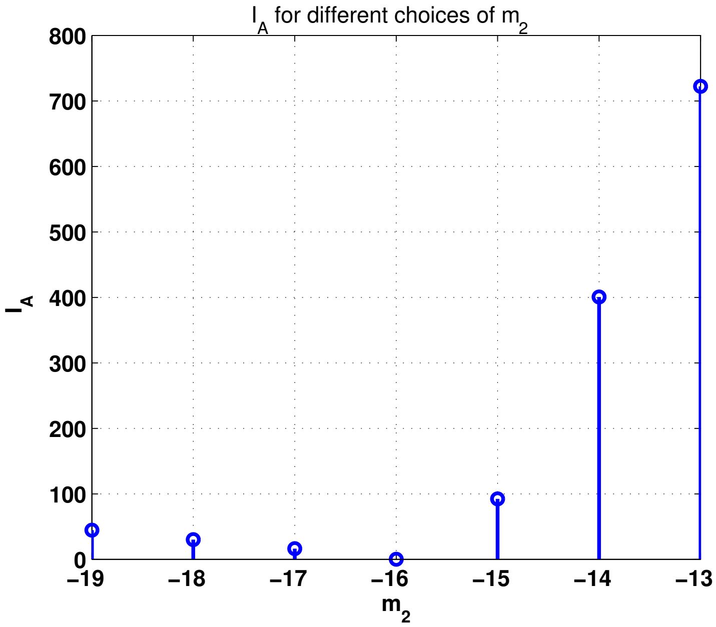

Figure 7.8: I_A for different choices of m_2

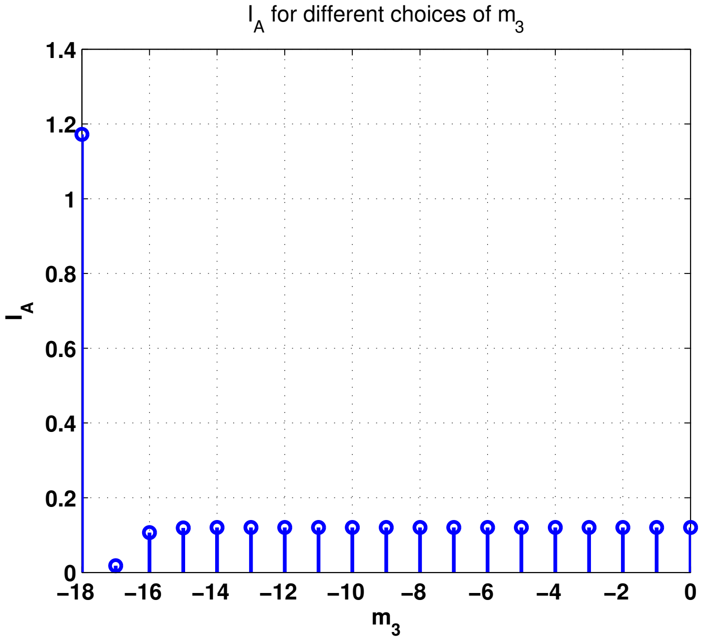

Figure 7.9: I_A for different choices of m_3

The optimal measurement noise variance is slightly higher then the variance used for data generation; this is most probably due to the fact that the filter and the data model are not exactly the same (different choice for the third state variable). Furthermore it is worthwhile to stress that the most important ratio is $\frac{q}{R}$; as a matter of fact from Figure 7.7 it can be seen that an improper choice of the measurement noise can increase I_A by factors of the order of 100, whereas the ratios between the process noise variances, especially the ratio $\frac{q_r}{q_h}$, are not that critical. Therefore the initial step is always to adjust $\frac{q}{R}$. Finally it can be stated that the tuning of the process noise variance ratios was not even necessary, since for values close to unity the output prediction error can be already considered to be perfectly white; more generally, even for values of I_A of the order of $20 - 30$ the prediction error can be considered as quasi-white noise.

The ACF and the PS for the optimal UKF tuning are reported in Figures 7.10 and 7.11; it is clear that the white noise hypothesis is satisfied. The ACF lies within the bounds, and the PS is indiscernible from the ideal white noise PS.

The sigma point parameters where not yet considered; however, no significant gain is obtained from their variation. This is not surprising, since with the standard choice we are already in white noise conditions. Once this situation is achieved, the filter has extracted all the useful information of the data and it is not possible to further improve its performance.

The procedure explained in this section was used for filter tuning throughout the whole project. However, as will be shown in the next chapter, for experimental data it is not possible to obtain such good results.

The estimated trajectories are reported in the next section, where the old UKF approach is compared to the improved UKF approach.

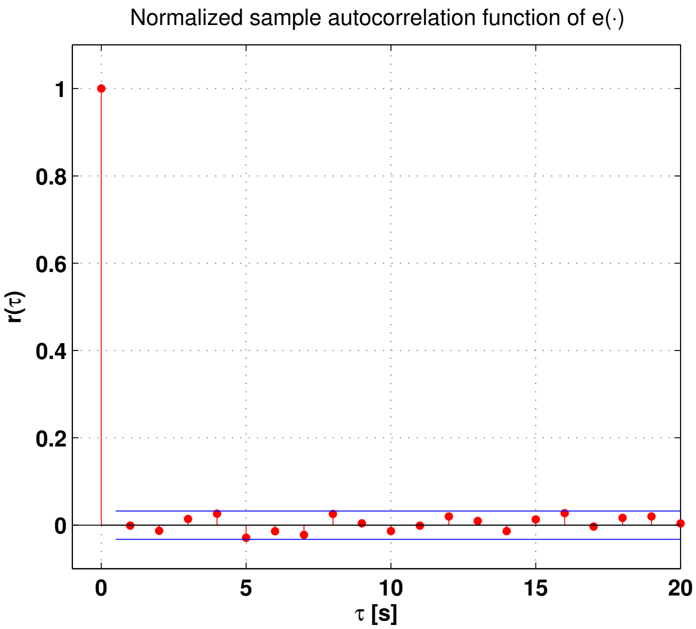

Figure 7.10: Normalized sample autocorrelation function of $e(\cdot)$

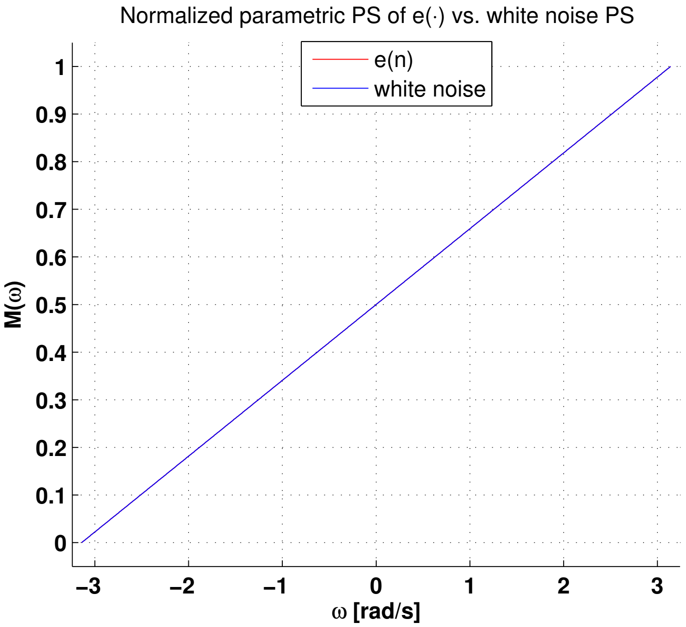

Figure 7.11: Normalized parametric PS of $e(\cdot)$ vs. white noise PS

7.2.2 Comparison between old UKF and improved UKF

The optimal tuning for the old and the improved UKF, obtained according to the procedure described in the previous section, yields approximately the same value for I_A (which is close to 0). The old UKF is the one used in [1], based on the original capillary model. The tuning was verified to be optimal considering the average over 10 random generated trajectories.

For both filters the standard sigma point parameters were used, since an adjustment of these values yields no significant improvement. I_A is slightly smaller for the improved UKF, however the difference is absolutely insignificant and the output prediction error is perfectly white in both cases. As will be pointed out later on, the principal advantage of the improved UKF is its major robustness with respect to variations of the model parameters.

Figures 7.12, 7.13, 7.14 and 7.15 report the comparison of the estimated crystal radius, meniscus height, growth rate and melt temperature. Note that the old UKF estimates the melt temperature, however its growth rate estimation can be computed readily applying formula (3.2) (and viceversa for the improved UKF; in this case the inverse of (3.2) is used).

It results that the radius estimation is approximately the same for both filters, whereas the difference is slightly more significant for the other variables. It turns out that for both the old and the improved UKF the best estimation of the radius and the meniscus height is quite irregular with respect to the true trajectories. This effect is even more pronounced for the experimental data, as will be shown in the next chapter.

Generally the estimation of the hidden state variables seems, at first appearance, quite poor. However this is nothing else than the proof of the fact that the filters have to be tuned according to a statistical quality index, i.e. the test for white noise, and not looking at the trajectories (moreover in the case of experimental data the true trajectories are not available, therefore this consideration becomes obsolete). As a matter of fact, we know that the chosen noise variances are optimal, whereas looking just at the trajectories they seem to be suboptimal.

It is worth mentioning that the considered filter tuning is optimal for state estimation, however this does not mean that it is optimal for control purposes too. As a matter of fact it could be that the estimations are too irregular for the LQR, causing undesired effects, such as oscillations, in the closed loop system. Such effects were observed by Siegert during his experiments on the real puller. Therefore it could happen that for the closed loop system the filter must be retuned in order to yield satisfying regulation results, at the expense of a no longer optimal state estimation. Some of these aspects will be considered in Section 7.3; however for meaningful conclusions on this problem the only way is to test the control system on a real puller, which was not possible so far.

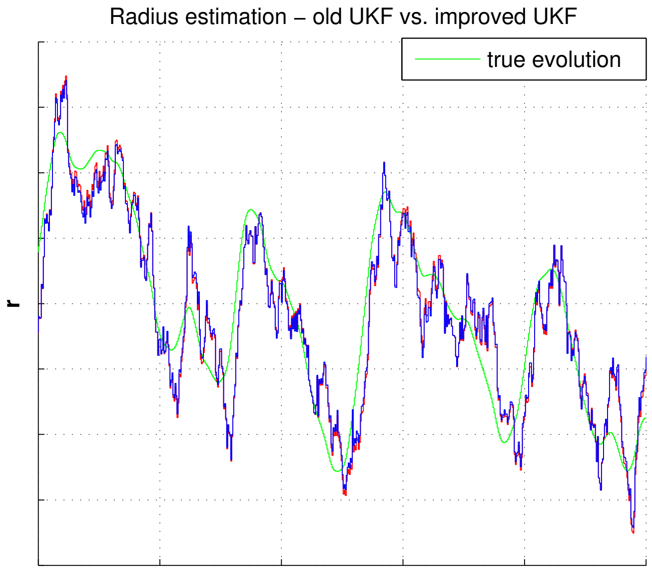

Figure 7.12: Radius estimation - old UKF vs. improved UKF

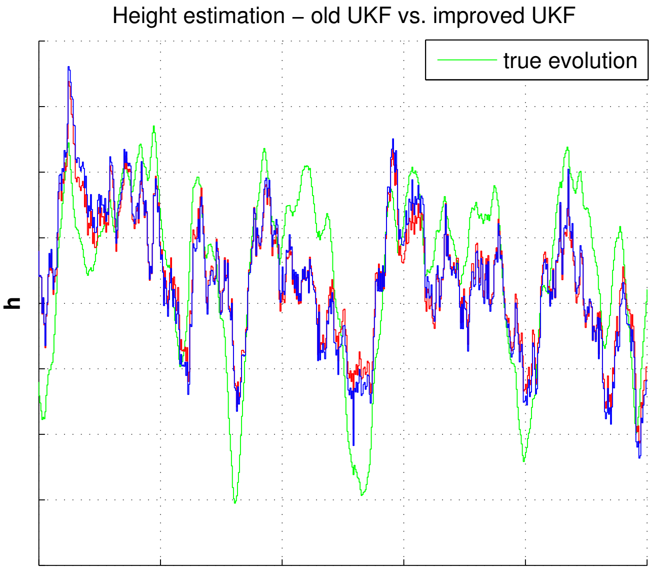

Figure 7.13: Height estimation - old UKF vs. improved UKF

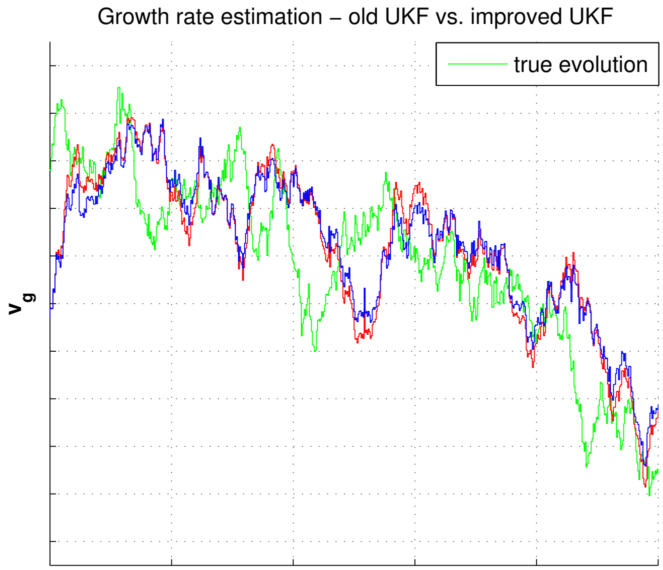

Figure 7.14: Growth rate estimation - old UKF vs. improved UKF

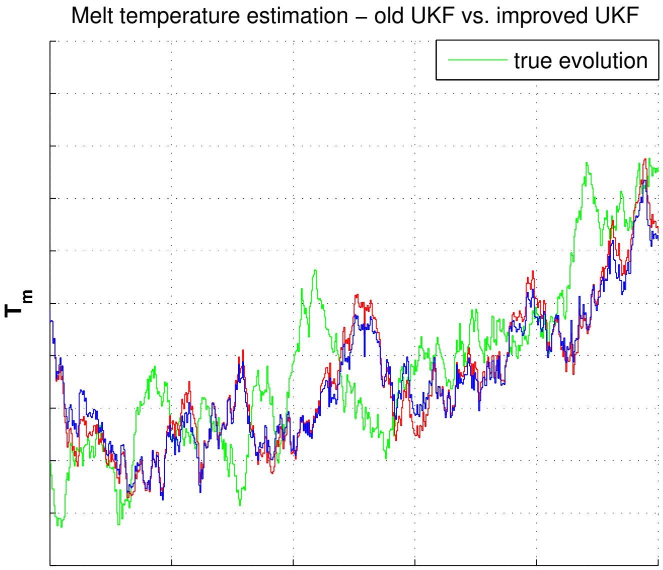

Figure 7.15: Melt temperature estimation - old UKF vs. improved UKF

Returning to the state estimation problem, the growth rate (and melt temperature) estimation shows an interesting phenomenon: there appears some kind of delay between the true trajectory and the estimated one, of the order of some seconds up to half a minute, for both the old and the improved UKF. This behaviour can be explained looking at the evolution of the output prediction error $e(n) = y(n) - \hat{y}(n|n-1)$. As it is well known, the Kalman Filter and its nonlinear extensions compute the state estimation from the state prediction, correcting it with a factor proportional to $e(n)$. However, if $e(n)$ is of the same order as the standard deviation of the measurement noise, that is

$$|e(n)| \cong \sqrt{R},$$

the filter attributes the error in the output prediction to the measurement noise; that is, it "believes" that the prediction was not wrong, and therefore it will not correct it. The estimation is adjusted only when the output prediction error exceeds the interval defined by the standard deviation of the measurement noise. This is proved by Figures 7.16 and 7.17. These Figures report, for the case of the improved UKF (however for the old UKF the principle remains the same), the first 30 seconds of the output prediction error and of the growth rate estimation. The red, dashdotted line corresponds to the interval defined by the standard deviation; at second 8 the prediction error abandons the interval for the first time, and from Figure 7.17 it can be seen that at the same instant of time the first (very small) adjustment on the growth rate estimation is made.[4] This behaviour is preserved throughout the entire duration of the simulation, explaining the time delay between true evolution and estimation. Depending on how fast $e(n)$ exceeds the bounds, the delay can become smaller or larger. From this qualitative discussion it turns out that reducing the measurement noise variance reduces the delay too. However by doing so the overall quality of the state estimation gets worse.

Finally, as stated at the beginning of this section, the variation of the sigma point parameters has no significant effect on the estimation performance. This suggests that the nonlinearities of the capillary model are not that "bad", that is they cause only a slight, negligible distortion of the state vector pdf. Obviously the unimodality of the state vector pdf is implied by the fact that the UKF works fine (as a matter of fact, for most cases of multimodal distributions it is not applicable). Furthermore the new insight on the sigma points leads to the hypothesis that most probably it is not too far away from a Gaussian distribution. This hypothesis is further encouraged by the comparison between the improved UKF and the EKF, reported in the next section. Finally it will be proved with the Particle Filter.

---

[4] actually, there appears an infinitesimal adjustment also on second 7; however approximately the consideration holds

Concluding Figure 7.18 reports the pdf of the output prediction error $e(n)$ for the case of the improved UKF (for the old UKF it is almost the same). The green graph shows the Gaussian pdf characterized by the sample mean and covariance of $e(n)$. From the graphical comparison it turns out that $e(n)$ is almost Gaussian distributed. Note that the histogram was computed, just like the white noise test, only on the portion of $e(n)$ with $n \geq 1200$, that is with a quite small number of samples; for this reason the pdfs look somewhat edgy.

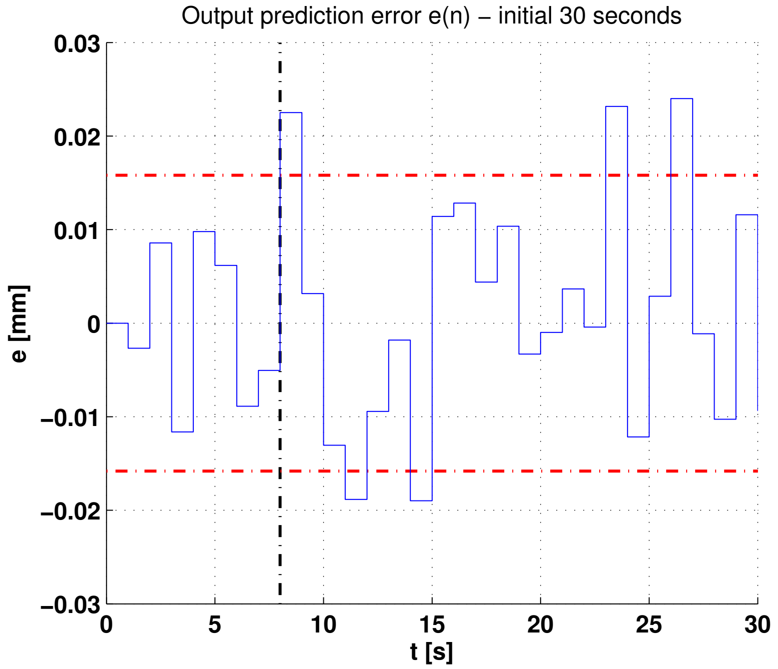

Figure 7.16: Zoom on the output prediction error $e(n)$ of the improved UKF

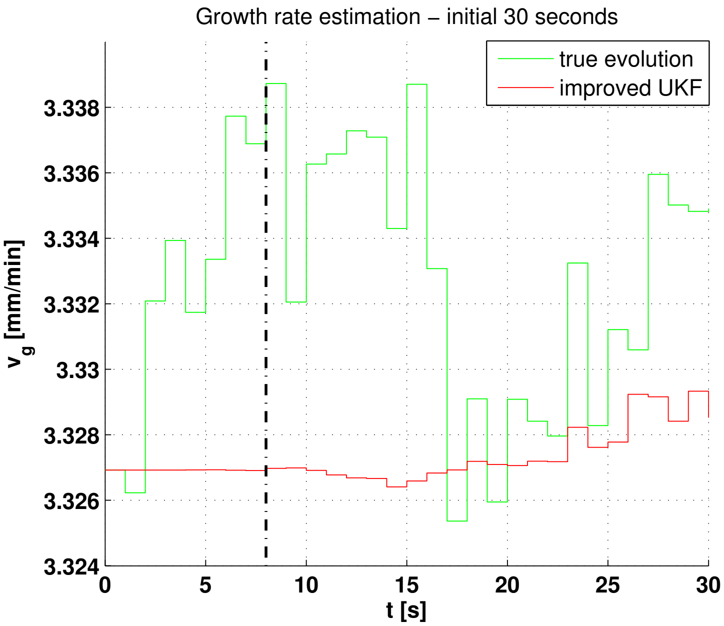

Figure 7.17: Zoom on the growth rate estimation of the improved UKF

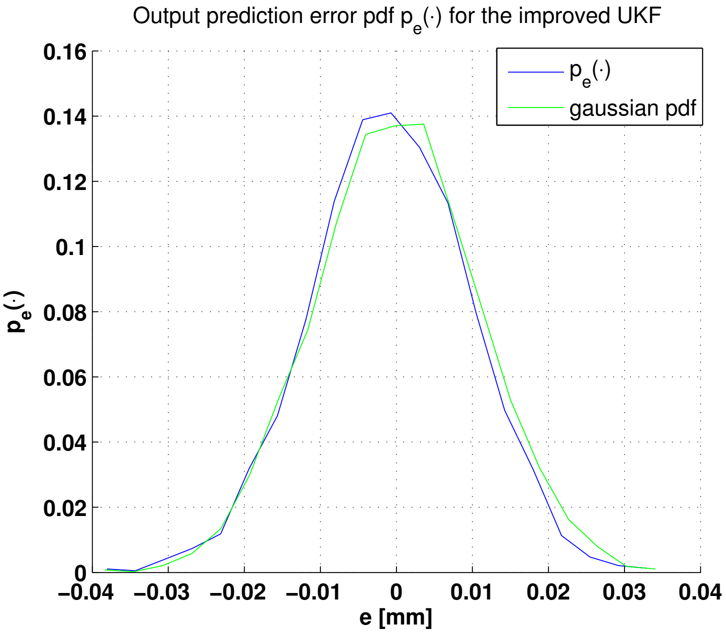

Figure 7.18: Output prediction error pdf $p_e(\cdot)$ for the improved UKF

7.2.3 Comparison between UKF and EKF

In the following the performance of the improved UKF is compared to the one of the EKF. The optimal tuning for these two filters yields approximately the same value for I_A (which furthermore is very small). A comparison of the radius and meniscus height estimation trajectories is illustrated in Figure 7.19 respectively 7.20. The optimal estimations of the improved UKF and the EKF are *exactly* the same, except for some small deviations on the meniscus height.[5] The same happens for v_g and T_m; for this reason the corresponding figures were omitted.

The optimal value of I_A for the EKF is slightly smaller than the one for the UKF, however the difference is, again, absolutely insignificant.

Therefore it turns out that the EKF performance is identical to the performance of the improved UKF; this is, at first appearance, quite surprising. As a matter of fact in Chapter 4 it was pointed out that usually the UKF yields a better performance. However, actually it is not that uncommon that the EKF behaviour is as good as the UKF behaviour; in fact often the nonlinear behaviour of the considered system is not that dramatic from the point of view of the state vector pdf distortion. As mentioned in the previous section, the capillary model seems to belong to this class of systems. This hypothesis is proved in the next section, applying the Particle Filter.

Obviously, since the EKF principle is simpler than the UKF principle, for equal performance it could be preferable to use the first one. Furthermore in the case of interest the EKF linearization relies one the same procedure as the linearization required for the LQR, and therefore the use of the EKF does not introduce any additional computation effort.

The distribution of the EKF output prediction error is very similar to the one of the improved UKF reported in Figure 7.18; for this reason it was omitted.

Finally it will be shown in the next chapter that the EKF performance is approximately equal to the UKF performance also in the case of experimental data.

---

[5] in fact the blue graph is covered by the red one

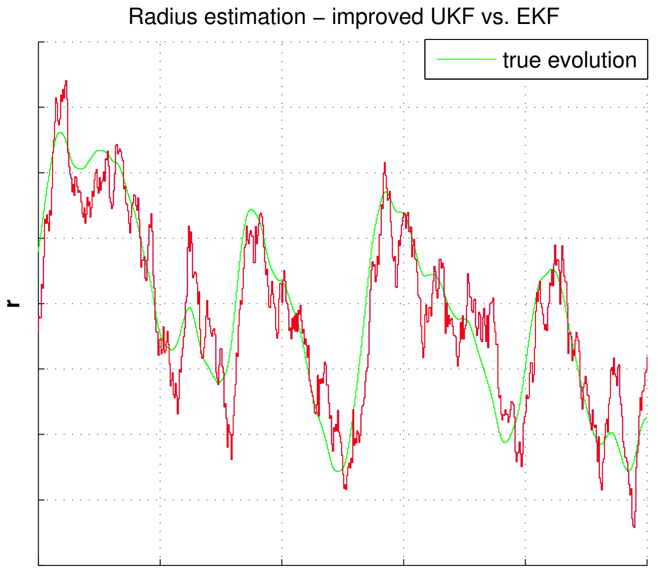

Figure 7.19: Radius estimation - improved UKF vs. EKF

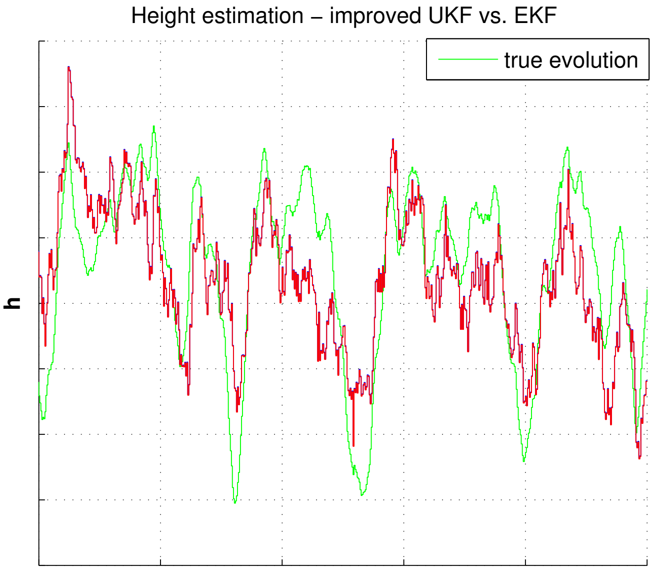

Figure 7.20: Height estimation - improved UKF vs. EKF

7.2.4 State vector pdf estimation with the Particle Filter

As explained in the previous sections, the behaviour of the EKF and the old and improved UKF leads to the conclusion that, despite the nonlinear dynamics of the capillary model, the state variables are almost Gaussian. In order to verify this hypothesis the Gaussian Particle Filter was applied to the simulated data. As stated in Chapter 4, for a large number of particles the PF approaches the *true* probability density function of the state vector.
The Particle Filter was initialized with the same $\hat{x}(0|0)$ and $P(0|0)$ as the other filters; furthermore it was parameterized with the optimal process noise covariance matrix and measurement noise variance of the improved UKF. For this choice the output prediction error computed by the PF is white and therefore its estimation is reliable. For the simulation $N_P = 1000$ particles were used, since with a larger number the computation becomes very expensive.
With respect to the time instant $n = 5000$ of the simulation, Figures 7.21, 7.22 and 7.23 report a comparison between the pdf estimation of the improved UKF and the Particle Filter. Furthermore, the Gaussian pdf characterized by the covariance and mean estimation of the PF is drawn as a reference. As explained in Chapter 4, the UKF approach approximates the true state vector pdf with a Gaussian one, characterized, at each time n, by the state estimation $\hat{x}(n|n)$ and the estimated covariance $P(n|n)$. Omitting the time dependence, and denoting the diagonal entries of $P(n|n)$ with p_{11}, p_{22} and p_{33}, whereas $\hat{x}_1$, $\hat{x}_2$ and $\hat{x}_3$ are the three elements of $\hat{x}(n|n)$, that is for the three state variables:

$$\begin{aligned}
p_r(r) &= \frac{1}{\sqrt{2\pi\, p_{11}}}\, exp\left(-\frac{1}{2}\frac{(r-\hat{x}_1)^2}{p_{11}}\right),\\
p_h(h) &= \frac{1}{\sqrt{2\pi\, p_{22}}}\, exp\left(-\frac{1}{2}\frac{(h-\hat{x}_2)^2}{p_{22}}\right),\\
p_{v_g}(v_g) &= \frac{1}{\sqrt{2\pi\, p_{33}}}\, exp\left(-\frac{1}{2}\frac{(v_g-\hat{x}_3)^2}{p_{33}}\right).
\end{aligned}$$

The red histograms were computed the same way, but using the PF estimation of the mean and the covariance (which are slightly different from the UKF estimation) for the parameterization of the Gaussian pdf. Like for the Particle Filter also the histograms for the Gaussian pdfs were computed on only 1000 samples, in order to get a meaningful comparison. Since the number of samples is quite reduced, the histograms have a very edged aspect.
The true pdfs are given by the Particle filter pdf estimations, that is the green graphs; from the Figures it is clear that they are almost Gaussian, since the are quite similar to the Gaussian pdfs computed with the Particle Filter mean and covariance. Therefore the UKF (and EKF) approach to pdf

approximation is very appropriate and precise. The histogram for the EKF is not reported since it is nearly identical to the one of the UKF.

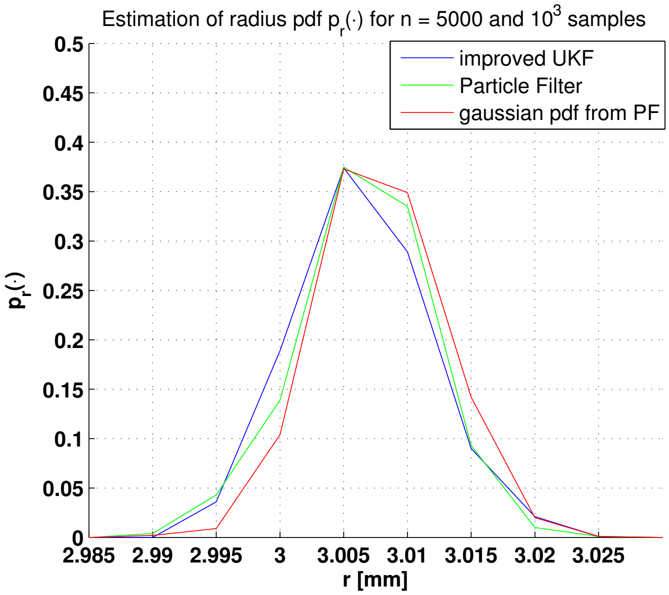

Figure 7.21: Estimation of radius pdf $p_r(\cdot)$ for $n = 5000$ and 10^3 samples

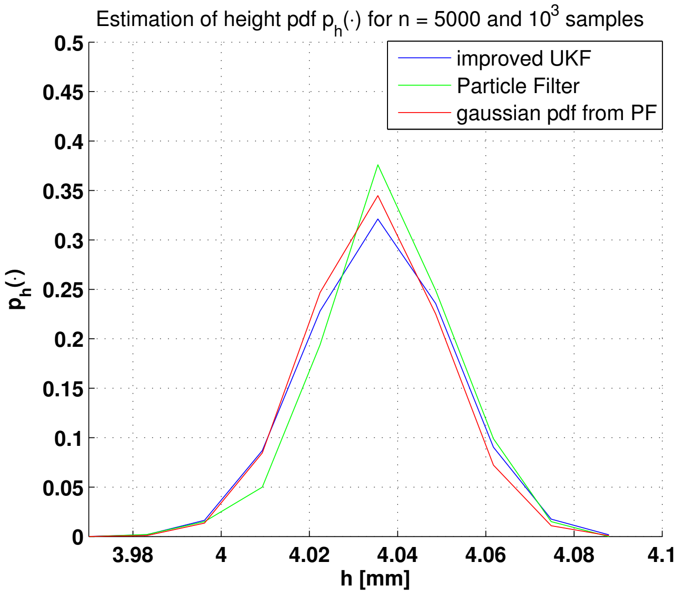

Figure 7.22: Estimation of height pdf $p_h(\cdot)$ for $n = 5000$ and 10^3 samples

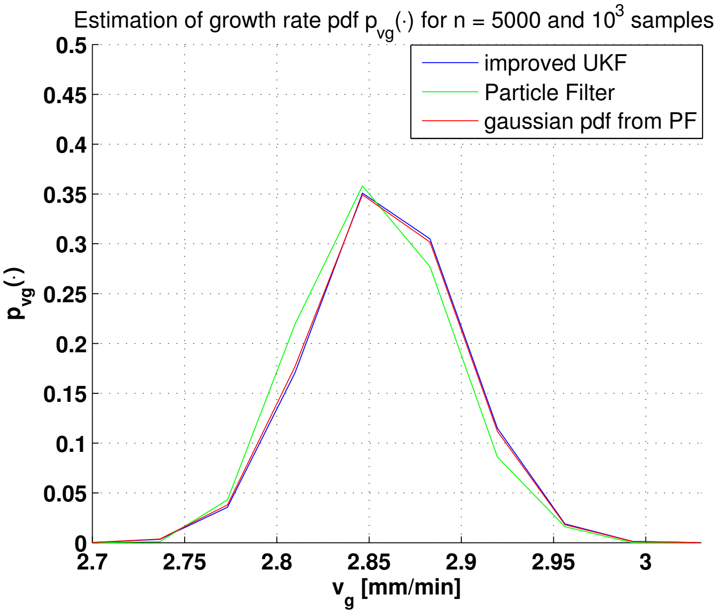

Figure 7.23: Estimation of growth rate pdf $p_{v_g}(\cdot)$ for $n = 5000$, 10^3 samples

The hypothesis of Gaussianity is therefore confirmed by the Particle Filter; however the pdfs were computed with a small number of samples, and therefore the representation is not that precise. In order to proof the Gaussian hypothesis, finally a one-shot prediction of the state variable pdfs is performed, with $N_p = 10^5$. For such a large number of samples the pdf representation is much more precise. It is sufficient to consider the prediction since this is the step which applies the nonlinear state update function; if the predicted pdfs are still Gaussian, as expected from the raw approximation described above, then our assumption is formally proved.
For the one-shot prediction the following $\hat{x}(n|n)$ and $P(n|n)$ were considered as the current state and state covariance estimation:[6]

$$\begin{aligned}\hat{x}(n|n) &= \begin{bmatrix} 3.150 \cdot 10^{-3} \\ 3.947 \cdot 10^{-3} \\ 6.832 \cdot 10^{-5} \end{bmatrix}, \\ P(n|n) &= P(0|0).\end{aligned}$$

Furthermore for the current input $u(n) = 4.20 \frac{mm}{min}$ was assumed.[7] Based on these current estimates, the one step ahead prediction of the state vector,

---

[6] they refer to the SI units; the exact numerical values are not important and were chosen randomly

[7] again, a randomly chosen value

$\hat{x}(n+1|n)$, and the covariance, $P(n+1|n)$, were performed with the EKF, the improved UKF and the PF respectively (obviously the PF computes the prediction of the entire pdf and not only of these two moments). As stated above, 10^5 samples were used for the histogram computations. Obviously the samples for the EKF and the UKF pdf prediction were drawn from the Gaussian pdfs reported above, parameterized with the predictions instead of the estimations. The predicted pdfs are reported in Figures 7.24, 7.25 and 7.26; note that the PF prediction was computed without resampling (and obviously without weight updating, since we are considering the prediction and not the corrected estimation). From the Figures it is crystal clear that the Gaussianity assumption made by the EKF and the UKF is justified for all three state variables; it is stunning how close the true pdfs computed by the PF are to the Gaussian pdfs assumed by the other filters.

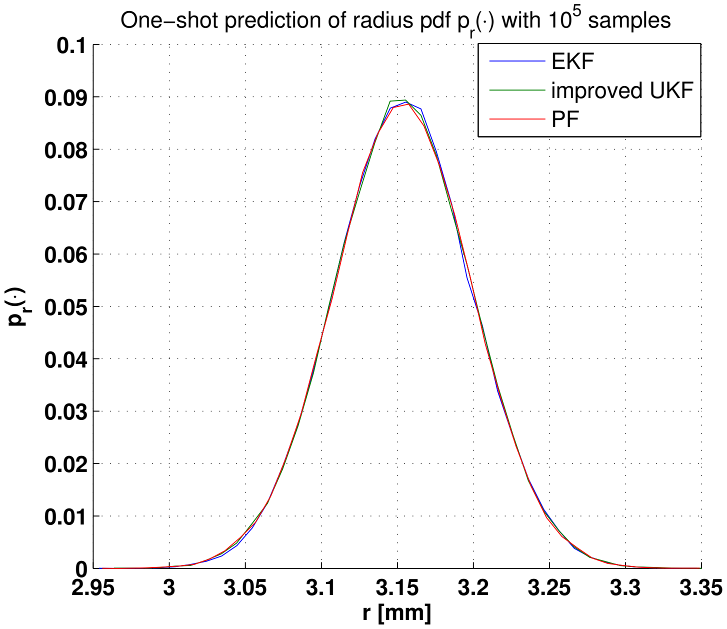

Figure 7.24: One-shot prediction of radius pdf $p_r(\cdot)$ with 10^5 samples

The exact numerical value of the prediction $\hat{x}(n+1|n)$ is not important in this context; however it is interesting to compare the predicted covariances $P(n+1|n)$ for the three cases. For a state space of dimension $n=2$ the graphical comparison of the covariances is straightforward, since the ellipse representation described in Section 4.6.1 is two-dimensional. Unfortunately in our case the state space has dimension $n=3$, yielding the ellipse representation of the covariances matrices 3-dimensional and therefore of difficult graphical interpretation. However the “distance“ between the co-

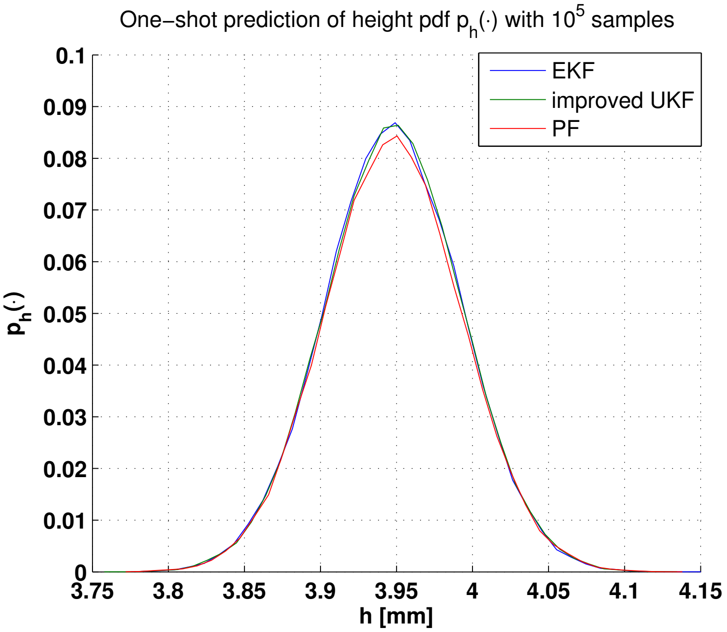

Figure 7.25: One-shot prediction of height pdf $p_h(\cdot)$ with 10^5 samples

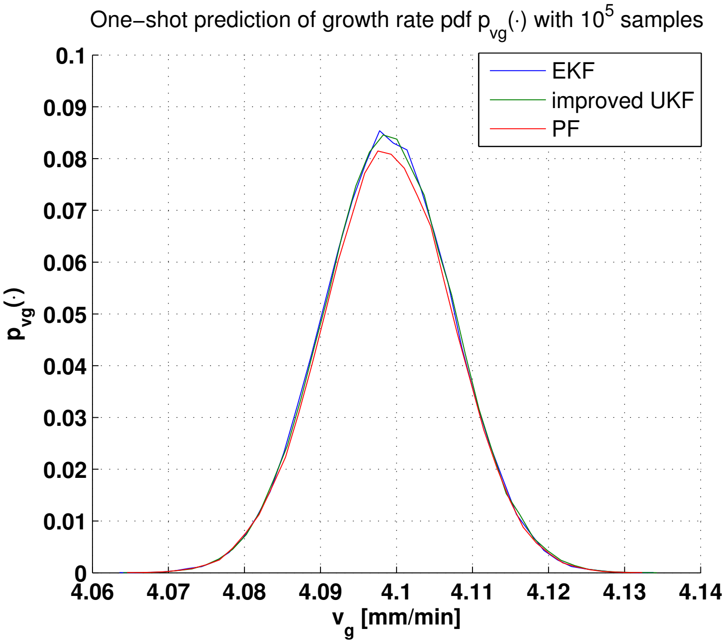

Figure 7.26: One-shot prediction of growth rate pdf $p_{v_g}(\cdot)$ with 10^5 samples

variance predictions made by the different filters can be quantified numerically resorting to the so-called Frobenius norm. Consider two positive-definite covariance matrices P and $\bar{P}$ contained in $\mathbb{R}^{3\times 3}$; as described in Section 4.6.1, the Eigenvalue Decomposition yields:

$$P = U\,\Sigma\,U^T = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} u_1^T \\ u_2^T \\ u_3^T \end{bmatrix},$$

$$\bar{P} = \bar{U}\,\bar{\Sigma}\,\bar{U}^T = \begin{bmatrix} \bar{u}_1 & \bar{u}_2 & \bar{u}_3 \end{bmatrix} \begin{bmatrix} \bar{\lambda}_1 & 0 & 0 \\ 0 & \bar{\lambda}_2 & 0 \\ 0 & 0 & \bar{\lambda}_3 \end{bmatrix} \begin{bmatrix} \bar{u}_1^T \\ \bar{u}_2^T \\ \bar{u}_3^T \end{bmatrix}.$$

The eigenvectors $u_1/\bar{u}_1$ define the orientation of the "x-axis", $u_2/\bar{u}_2$ the orientation of the "y-axis" and $u_3/\bar{u}_3$ the orientation of the "z-axis" of the 3-dimensional ellipse associated with the matrices.[8] On the other hand the square roots of the corresponding eigenvalues $\lambda_i/\bar{\lambda}_i$, $i = 1, 2, 3$, define the length of the ellipse along these axes. The matrices U and $\bar{U}$ are orthonormal; that is:

$$U^T U = \bar{U}^T \bar{U} = I.$$

I is the identity matrix. If the orientation of the axes associated with P and $\bar{P}$ is exactly the same it holds that

$$U^T \bar{U} = I$$

too. The deviation between the axes orientation of P and $\bar{P}$ can be quantified with the Frobenius norm $||\cdot||_F$. For a matrix M it is defined as:[9]

$$||M||_F = \sqrt{\mathrm{Tr}(MM^T)} \in \mathbb{R}.$$

The operator $Tr(\cdot)$ denotes the trace of the matrix, that is the sum of its diagonal entries. Applying this norm to the matrix $I - U^T\bar{U}$ the orientation deviation between the axes of P and $\bar{P}$ is quantified; furthermore, applying it to $\Sigma - \bar{\Sigma}$, the difference between the length over these axes is computed. More precisely, the following quantities are considered:

$$\begin{aligned} ||I - U^T\bar{U}||_F &= \sqrt{\mathrm{Tr}((I - U^T\bar{U})(I - U^T\bar{U})^T)}, \\ ||\Sigma - \bar{\Sigma}||_F &= \sqrt{\mathrm{Tr}((\Sigma - \bar{\Sigma})(\Sigma - \bar{\Sigma})^T)}. \end{aligned}$$

---

[8] the terms x-axis, y-axis and z-axis are not entirely correct and were adopted only for illustrative reasons

[9] in Matlab, the Frobenius norm of M is computed with the instruction *norm(M, 'fro')*

Lets now consider the covariance matrices predicted by the EKF, the improved UKF and the PF; in the case of interest, and reporting only the diagonal entries, they are given by:

$$\begin{aligned}
P_{EKF}(n+1|n) &= \begin{bmatrix} 2.025 \cdot 10^{-9} & * & * \\ * & 2.000 \cdot 10^{-9} & * \\ * & * & 2.000 \cdot 10^{-14} \end{bmatrix}, \\
P_{UKF}(n+1|n) &= \begin{bmatrix} 2.025 \cdot 10^{-9} & * & * \\ * & 2.000 \cdot 10^{-9} & * \\ * & * & 2.000 \cdot 10^{-14} \end{bmatrix}, \\
P_{PF}(n+1|n) &= \begin{bmatrix} 2.041 \cdot 10^{-9} & * & * \\ * & 2.006 \cdot 10^{-9} & * \\ * & * & 2.000 \cdot 10^{-14} \end{bmatrix}.
\end{aligned}$$

The diagonal entries of the EKF and UKF covariances differ only after the tenth decimal place ca. Furthermore these entries, which are the scalar variances of the radius, the meniscus height and the growth rate respectively, confirm the theoretical considerations exposed in Chapter 4: typically EKF and UKF tend to underestimate the state estimation/prediction error variances. In fact their predictions are smaller than the true variances given by the PF (except for v_g).

Finally the "deviations" between the covariance ellipses are computed with the Frobenius norm, first between the EKF and the UKF ellipse, and then between the UKF and the true PF ellipse:[10]

- Deviation between the axes orientation for EKF and UKF:

  $$||I - U_{EKF}^T\, U_{UKF}||_F = 5.482 \cdot 10^{-7}.$$

  Deviation between the axes length for EKF and UKF:

  $$||\Sigma_{EKF} - \Sigma_{UKF}||_F = 3.893 \cdot 10^{-17}.$$

- Deviation between the axes orientation for UKF and PF:

  $$||I - U_{UKF}^T\, U_{PF}||_F = 7.356 \cdot 10^{-2}.$$

---

[10] for EKF-PF the values are almost the same as for the UKF-PF comparison, they differ only on insignificant decimal places

Deviation between the axes length for UKF and PF:

$$||\Sigma_{UKF} - \Sigma_{PF}||_F = 1.690 \cdot 10^{-11}.$$

Although the interpretation of these values is difficult, they confirm that the ellipses computed by the EKF and UKF are nearly identical, whereas the deviation with respect to the PF ellipse is more significant. Anyway the deviation is not dramatic and therefore the covariance estimations and predictions of the EKF and UKF are reliable.

7.2.5 Filtering robustness discussion

As pointed out previously, the old UKF yields approximately the same estimation performance as the improved UKF and the EKF, if tuned correctly. However this does not mean that the application of the improved capillary model for state estimation (and radius regulation) is useless. As a matter of fact a delicate point was not yet discussed, that is the robustness of the estimation (and of the regulation) with respect to model parameter variations. From this point of view the new approach is by far superior. The model parameters were discussed in Section 3.2.2 and are recalled below; the bold parameters are the ones which compare also in the improved capillary model.

Model parameter	***Symbol***
Melting point temperature of silicon	T_c
Thermal conductivity of solid silicon	λ_s
Thermal conductivity of liquid silicon	λ_L
Density of solid silicon at T_c	ρ_s
Density of liquid silicon at T_c	ρ_m
Specific latent heat of fusion	ΔH_f
Capillary constant	**a**
Equilibrium growth angle	$\boldsymbol{\theta_C}$
Temperature gradient over the crystal	G_s

Table 7.1: Model parameters

In Section 3.2.2 it was pointed out that most of the parameters are affected by huge uncertainties. With the approach adopted in [1] the estimation, as well as the regulation (LQR), depends on each of these parameters, and therefore it is very sensitive to the chosen values. A wrong choice (that is, for simulation, a mismatch between the values used for the filter and the data generation model) can have critical effects on the estimation quality. Furthermore an inappropriate choice of most of these parameters affects only the estimation of the hidden state variables, especially the melt temperature, whereas the radius estimation is typically deviated only in the initial, transitory phase. After the transitory the radius estimation returns to the "correct" value, as well as the radius prediction. This means that an inappropriate choice of one or more model parameters has not a big impact on the white noise index, which is based on the radius prediction. Working on simulated data this is not that problematic, since we can compare the estimated trajectories to the true ones and state whether they are (approximately) correct or not; but on experimental data there is no way to make such evaluation, since the true evolution of the hidden state variables is unknown in this case. In order to understand whether the parameters were chosen appropriately or not the only way is to test the closed loop system on the real puller; if the control loop yields poor results, they have to be adjusted with a trial-and-error procedure, starting from the nominal values. This can be very tricky.

In order to illustrate the situation (for the old UKF), the effect of a 10% perturbation of the temperature gradient G_s, which is one of the most uncertain parameters,[11] is reported in Figures 7.27, 7.28, 7.29 and 7.30. As stated above, apart from the initial transitory, practically nothing changes for the radius estimation (as well as for the radius prediction, not shown in the Figures). Furthermore the area index has changed only slightly; this is not surprising, since it does not take into account the initial part of the prediction error. However from the Figures it is clear that the melt temperature estimation is wrong; there appears a constant offset of about $2\ ^\circ C$. Basically the filter *adjusts* the melt temperature estimation in order to obtain a correct radius (and meniscus height) estimation. As can be seen from Figure 7.30, the growth rate estimation, which is *not* computed directly by the old UKF approach, assumes the correct value after the transitory, and therefore the radius and the meniscus height estimation too. However this is not entirely correct since the estimated variable used by the original LQR (for the computation of the Jacobian matrices) is the melt temperature and not the growth rate, and therefore the regulator relies on wrong state vector estimations.

---

[11] the mismatching was achieved varying the parameter in the filter model and not in the data generation model, in order to rely on the same reference data as before

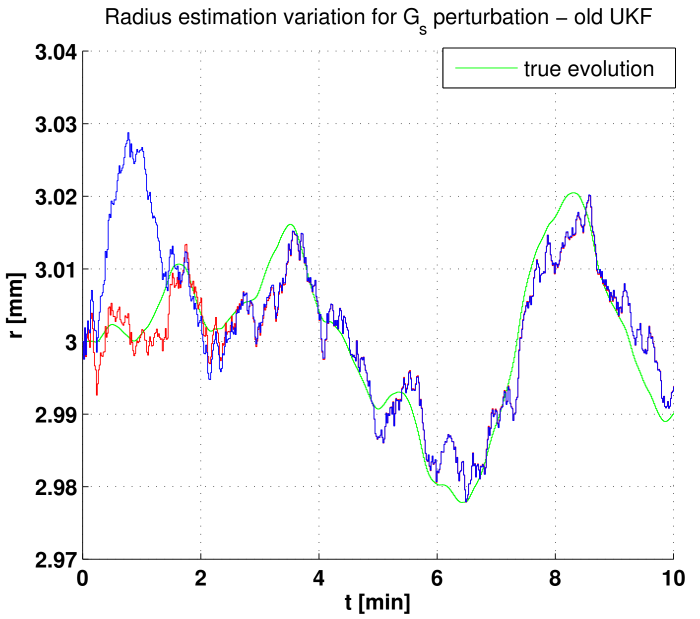

Figure 7.27: Radius estimation variation for G_s perturbation - old UKF

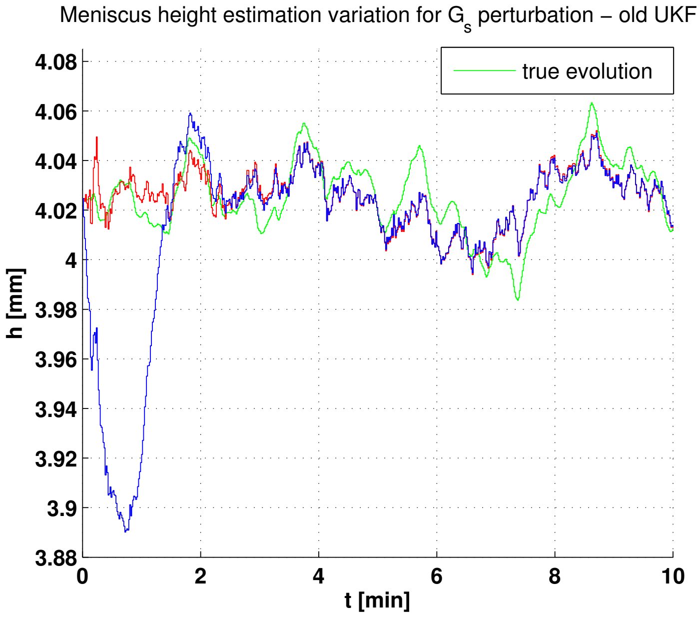

Figure 7.28: Meniscus height estimation variation for G_s perturbation - old UKF

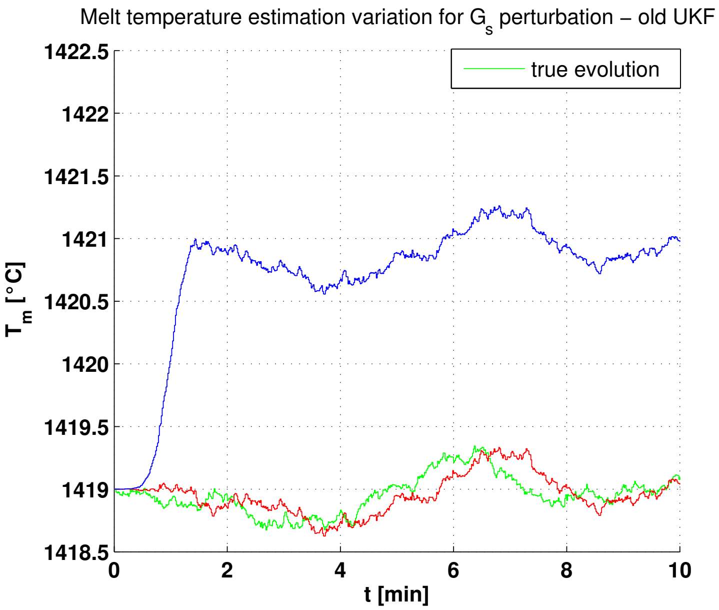

Figure 7.29: Melt temperature estimation variation for G_s perturbation - old UKF

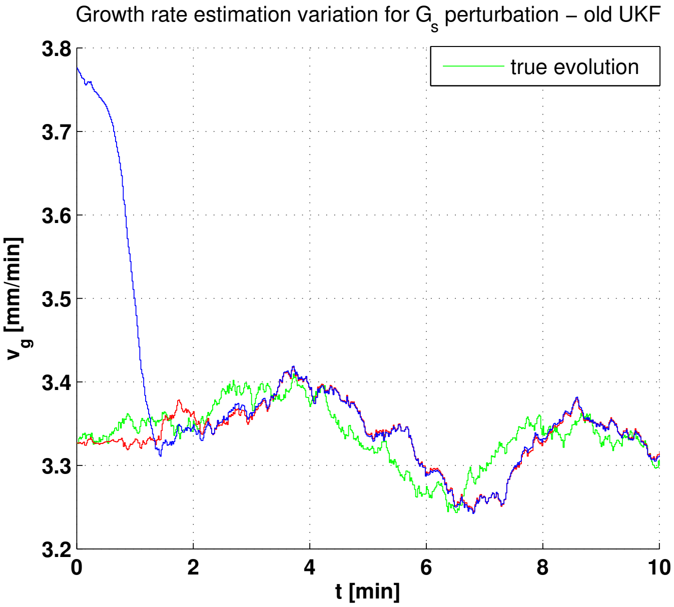

Figure 7.30: Growth rate estimation variation for G_s perturbation - old UKF

Analogous considerations hold for the other parameters; for this reason a detailed analysis was omitted. It is worth mentioning that the only two parameters that heavily affect also the meniscus height estimation (causing a constant offset not only on T_m, but also on h) are the capillary constant a and the equilibrium growth angle θ_C. This is consistent with the equations of the original capillary model, since these are the only values which affect the relation between the radius and the meniscus height (through the meniscus height approximation formula). Therefore, in order to maintain the radius estimation correct, the filter has to adjust the meniscus height estimation too. The other parameters are involved only in the growth rate computation and therefore also the height estimation can be kept at the correct value by simply shifting the melt temperature estimation.

Using the improved capillary model for both the state estimation (as done for the improved UKF, the EKF and the PF) and the LQR, the total quantity of model parameters is drastically reduced. As a matter of fact, only the capillary constant a and the equilibrium growth angle θ_C are left in the improved model, since all the other parameters are involved only in the growth rate formula which is no longer used. This is a huge advantage, since a lot of uncertainty is eliminated this way and therefore the robustness is increased significantly. Furthermore a and θ_C are known with quite good approximation, which further improves the robustness.

Anyway, robustness considerations should refer to the interconnection between the state estimator and the controller, that is to the closed loop system. Furthermore in order to obtain an idea of the robustness of the designed system it has to be applied to the real puller, which would be the next step in order to continue the work started with this project. It is there where it turns out how sensitive the system is to the model parameter values.

7.3 LQR simulations

The discussion of the simulation results is concluded with some considerations on the reduced state space LQR approach. For this purpose the optimally tuned UKF based on the improved capillary model was considered for state estimation. However the results are perfectly compatible with EKF state estimation since its performance is almost the same, as discussed earlier. The simulations were performed with the closed-loop Simulink scheme reported in Figure 7.1.
In the following the radius and meniscus height evolutions reported in the figures refer to the estimated trajectories and not to the true ones (generated by the model). This convention was adopted for compatibility with the experimental case, where these are the only available values.

First the effects of Bryson's rule are shown. Considering the set point[12]

$$x_{set} = \begin{bmatrix} r_{set} \\ h_{set} \end{bmatrix} = \begin{bmatrix} 3\ mm \\ 4.025\ mm \end{bmatrix},$$

where h_{set} was computed in order to obtain $\theta = \theta_C$ and therefore constant radius growth, two different penalty sets based on Bryson's rule were used:

- First choice
  Bryson's penalties designed for a maximum allowed radius and meniscus height deviation of 0.03 mm and a maximum allowed pull rate variation of 0.3 $\frac{mm}{min}$, that is:

$$Q_{LQR} = \begin{bmatrix} 1.\dot{1} & 0 \\ 0 & 1.\dot{1} \end{bmatrix}, \qquad R_{LQR} = 40, \qquad \Delta v_{max} = -\Delta v_{min} = 0.3\,\frac{mm}{min}.$$

- Second choice
  Bryson's penalties designed for a maximum allowed radius and meniscus height deviation of 0.02 mm and a maximum allowed pull rate variation of 0.5 $\frac{mm}{min}$, that is:

$$Q_{LQR} = \begin{bmatrix} 2.5 & 0 \\ 0 & 2.5 \end{bmatrix}, \qquad R_{LQR} = 14.4, \qquad \Delta v_{max} = -\Delta v_{min} = 0.5\,\frac{mm}{min}.$$

Figures 7.31 and 7.32 report the comparisons of the radius and the height evolutions, whereas Figures 7.33 and 7.34 show the pull rate variations for

---

[12] and using $x_0 = [x_{set}^T\ 1419°C]^T$ for the capillary state space model

the two cases. Note that only the first 10 minutes of the simulation are displayed, in order to improve the readability.
From the radius evolutions it turns out that the limits used for the penalty computation are satisfied; in both cases the difference $r - r_{set}$ does not exceed the pre-defined boundaries (± 0.03 mm respectively ± 0.02 mm around the set point). Furthermore in the first case this difference is allowed to assume slightly larger values than in the second case, as expected.
The effect is not that significant for the meniscus height; however, also in this case it can be observed that the distance from the set point is reduced for the second choice. Anyway for the latter choice of the penalties the specifications on $h - h_{set}$ are not satisfied. This is not that surprising since our system is nonlinear and furthermore the LQR relies on the estimated and not the true values of the state variables; thus the LQR approach yields only suboptimal performance.
Finally, the specifications on the pull rate variation are matched very well, as can be seen from Figures 7.33 and 7.34.
Therefore Bryson's rule is a valuable approach to penalty design also for our nonlinear case, since we are able to approximately satisfy the desired specifications with this method. Obviously on the real puller the performance will be not that good; it might be that for the experimental case the limits are not respected. This has to be figured out testing the LQR on the puller.

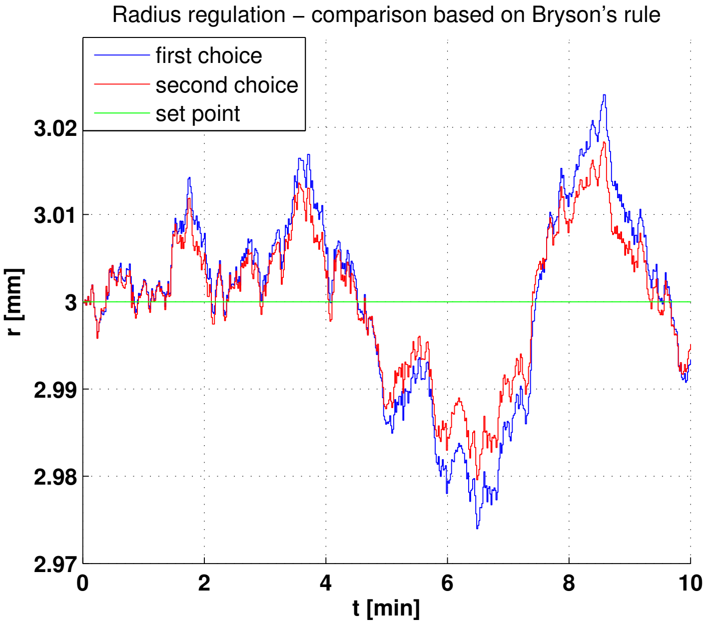

Figure 7.31: Radius regulation - comparison based on Bryson's rule

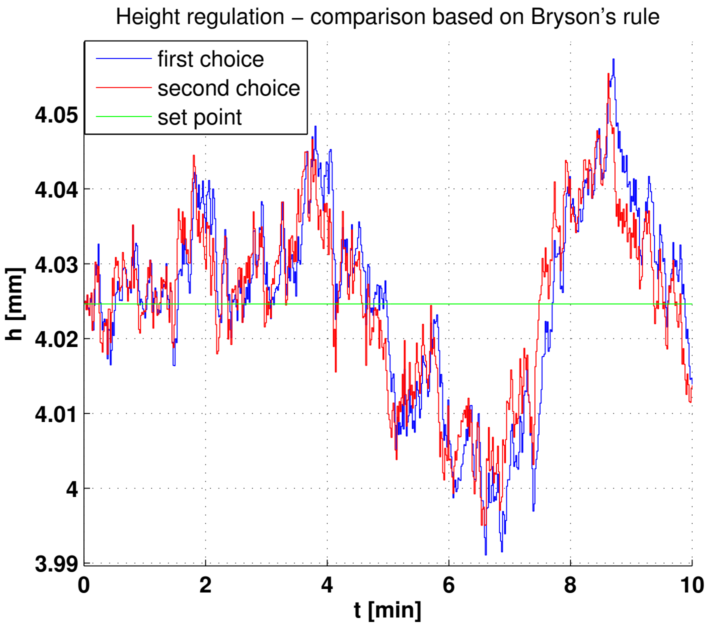

Figure 7.32: Height regulation - comparison based on Bryson's rule

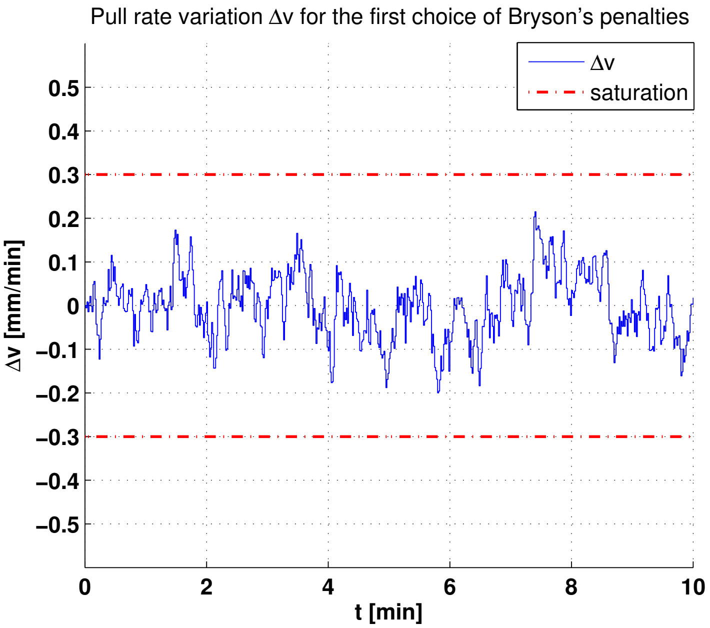

Figure 7.33: Pull rate variation Δv for the first choice of Bryson's penalties

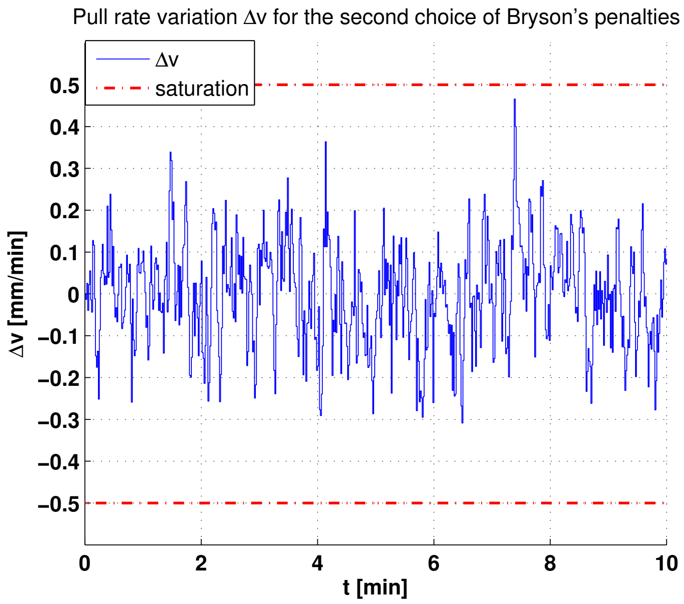

Figure 7.34: Pull rate variation Δv for the second choice of Bryson's penalties

The next step consists in the verification of the considerations reported in Section 6.2. For this purpose the state penalty matrix is factorized as follows:[13]

$$Q_{LQR} = \alpha \begin{bmatrix} q_r & 0 \\ 0 & q_h \end{bmatrix}.$$

Initially q_r and q_h are set to 1 and only the ratio

$$\frac{\alpha}{R_{LQR}}$$

is considered. For the following comparisons the saturation on the pull rate variation was set to $\pm\, 0.5\frac{mm}{min}$.

Figures 7.35 and 7.36 report the comparison of the radius and meniscus height regulation for different choices of R_{LQR}, keeping $\alpha = 1$. The shown time interval (from minute 78 to minute 83) was chosen in order to pick trajectory parts for which the effect of the ratio variation is conspicuous. The set point is represented in green, as usual.

---

[13] here the notation q_r/q_h is used for the radius/height penalties; it is the same notation adopted for the noise variances. Since in this section only the LQR is discussed, this should not create confusion, and therefore the subscript "LQR" was omitted, in order to simplify the notation. Finally R_{LQR} is the penalty on the pull rate variation

Figures 7.37, 7.38 and 7.39 show the pull rate variations for decreasing ratios. The trajectories are reported in different figures since they are difficult to distinguish when overlapped.
The Figures confirm the considerations reported in Section 6.2; increasing R_{LQR} with respect to α the pull rate variation becomes more expensive (or, equivalently, the state vector deviation from the set point becomes cheaper) and therefore the LQR tends to reduce the control signal, at the expense of a worse state regulation. This turns out for both the crystal radius and the meniscus height, as expected. However, even for a very large input penalty the deviation remains small for both the radius and the meniscus height, which is a quite good result. Anyway these simulation results are ideal and the oscillations of the regulated state variables around the set point will be by far higher on the puller.

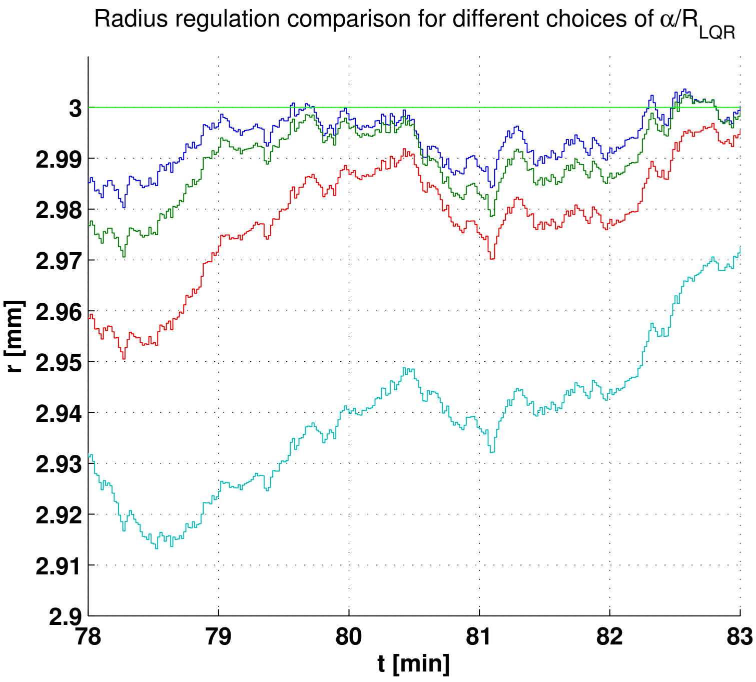

Figure 7.35: Radius regulation comparison for different choices of $\frac{\alpha}{R_{LQR}}$

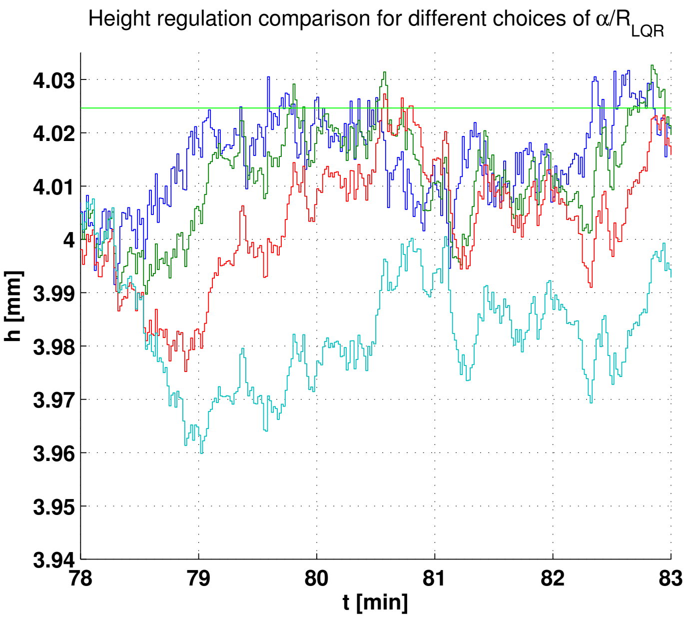

Figure 7.36: Height regulation comparison for different choices of $\frac{\alpha}{R_{LQR}}$

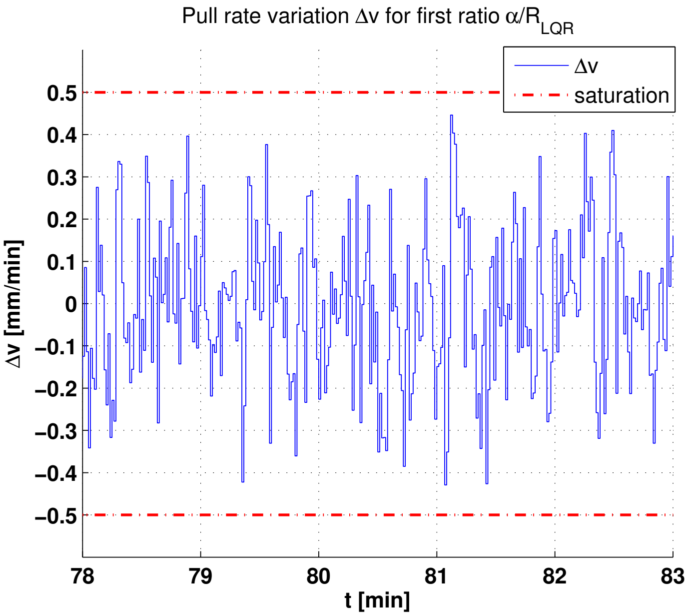

Figure 7.37: Pull rate variation Δv for first ratio $\frac{\alpha}{R_{LQR}}$

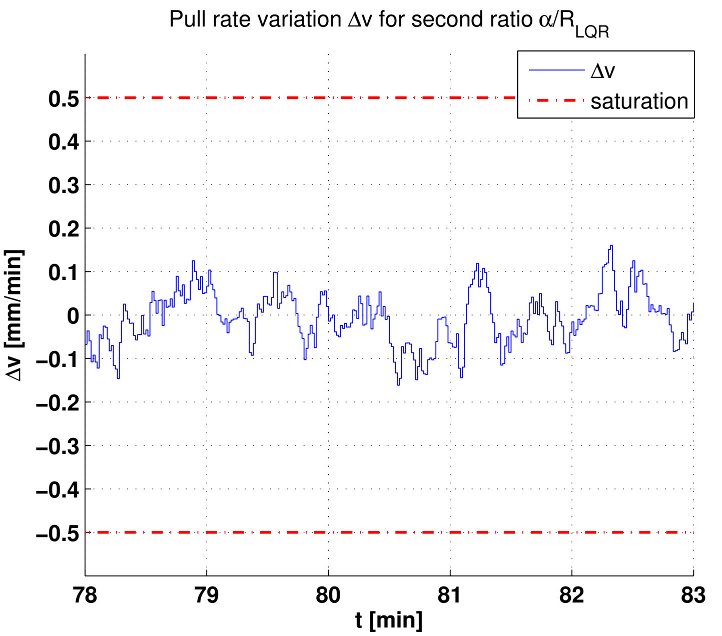

Figure 7.38: Pull rate variation Δv for second ratio $\frac{\alpha}{R_{LQR}}$

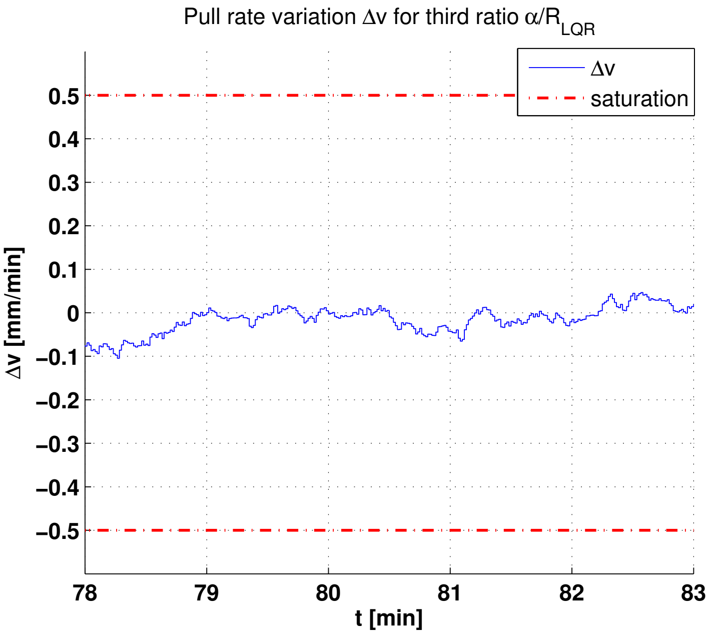

Figure 7.39: Pull rate variation Δv for third ratio $\frac{\alpha}{R_{LQR}}$

Finally different choices for the ratio

$$\frac{q_r}{q_h}$$

are considered. Figures 7.40 and 7.41 report the comparisons of the radius and the meniscus height regulation for 4 different choices of the ratio. Again, only a small part of the overall trajectories is shown, in order to improve the readability; furthermore the set points are given by the green lines. Considering first Figure 7.41, which compares the meniscus height trajectories, it turns out that with increasing penalty on h the deviation from the set point becomes smaller, as it should be. However the results shown in Figure 7.40, regarding the radius, are quite surprising; also the radius regulation improves as q_h is increased with respect to q_r (that is, equivalently, q_r is reduced with respect to q_h, since we are considering the ratio), whereas we expect it to worsen. This phenomenon was already predicted in Section 6.2, where the penalty choices were discussed. Now it is confirmed by the simulation results; due to the particular nonlinear relation between the crystal radius and the meniscus height, defined by the Johansen-Boucher formula (or the Boucher or the Tsivinskii formula), generally the state variables can not be regulated indipendently; if one of the variables is regulated closer to the set point, also the other one is regulated better, and vice versa.

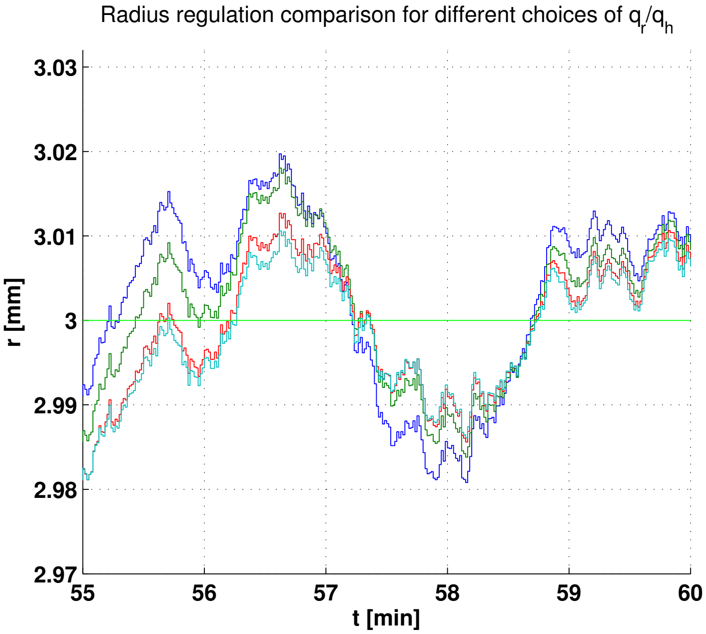

Figure 7.40: Radius regulation comparison for different choices of $\frac{q_r}{q_h}$

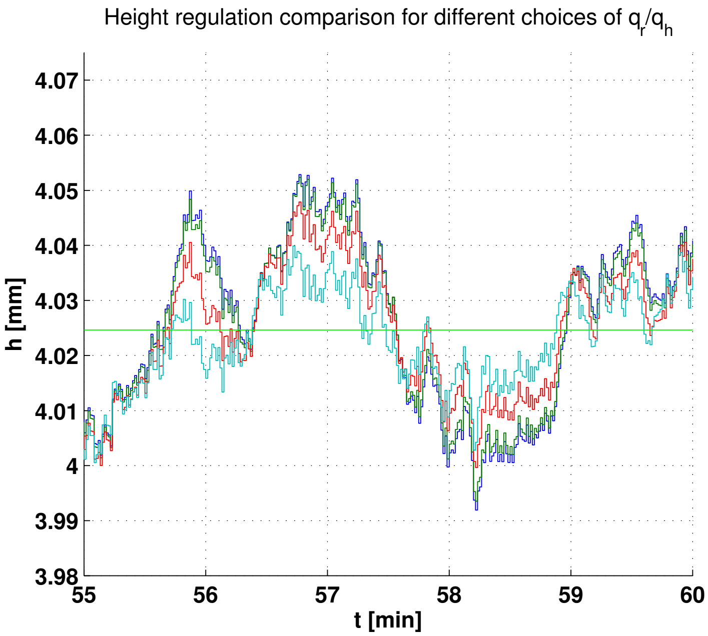

Figure 7.41: Height regulation comparison for different choices of $\frac{q_r}{q_h}$

Finally it is worth mentioning that the "oscillations" of the radius and of the meniscus height around the set point are caused by the growth rate, which acts as an external disturbance and is only approximately known to the LQR, since it has to rely on the estimation provided by the state estimator (the improved UKF for the simulations described in this section). This is confirmed by Figures 7.42 and 7.43, which compare the radius and height trajectory shape to the shape of the growth rate; it turns out that they are very similar. If the radius, meniscus height and growth rate trajectories were known *exactly* to the LQR, the radius and the meniscus height regulation would be perfect, that is r and h would *exactly* coincide with the set points (since we considered the system to be initially in equilibrium conditions). However on a real system this can not be achieved since we always have to rely on estimations. Therefore the filter tuning, for the tests of the closed loop system on an actual puller, has to be considered together with the LQR performance since it is conditioned by the quality of the estimations. It might be that the best statistical choice for the filter parameters (that is, its noise covariances) causes inacceptable regulation behaviour. Anyway such considerations can be made only on the real system; the simulation results are practically worthless for this purpose.

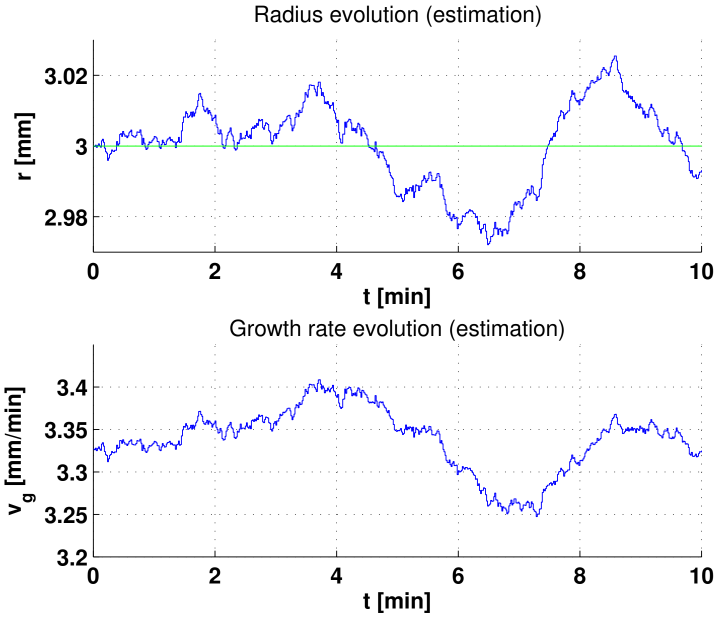

Figure 7.42: Comparison of radius and growth rate evolution

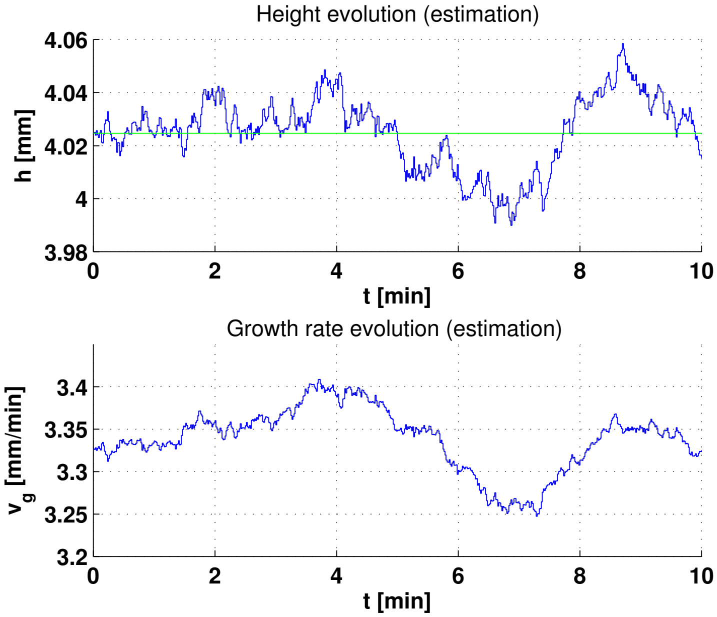

Figure 7.43: Comparison of height and growth rate evolution

Finally Figure 7.44 reports an example for radius tracking as it is required for necking. As a matter of fact usually the initial radius is larger than the desired neck radius and therefore the crystal has to be tapered, according to a pre-defined mathematical function. In the Figure tapering is obtained tracking just an inclined line; on the actual system typically a parabolic function is used. The reduced state space LQR works fine also for such tracking problems, as shown in the Figure (the set point is given by the blue line).

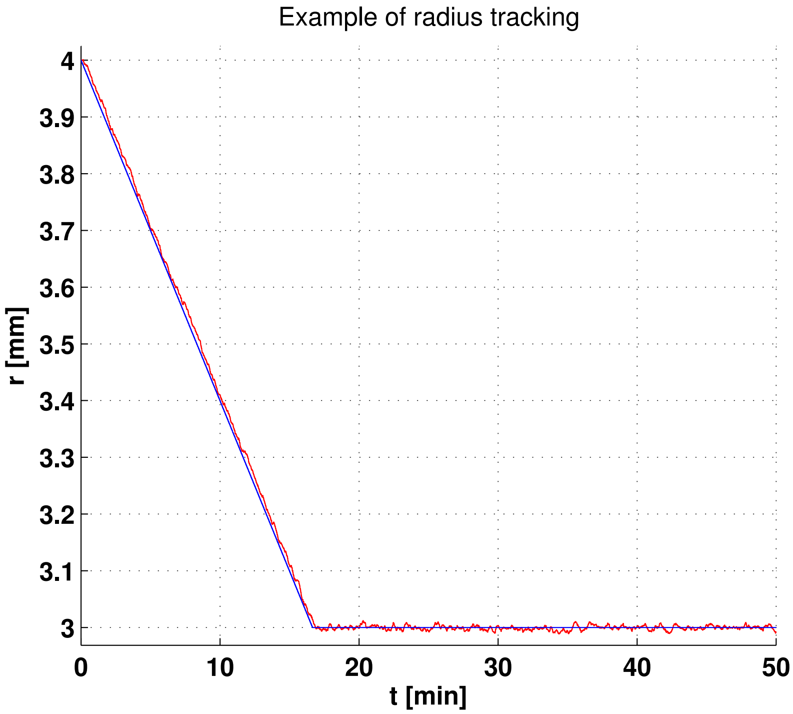

Figure 7.44: Example of radius tracking

Chapter 8

Filtering of experimental data

In this chapter the filtering algorithms - more precisely the old UKF, the improved UKF and the EKF - are applied to experimental data.[1] The data was provided by Siegert from MEMC St. Peters; it was recorded during his experiments with the closed loop system, composed by the original LQR and UKF, on a test puller in St. Peters. The results reported in this chapter refer to a particular data set, namely the first hour of the trajectories of the logfile *cplqr-20100311T******.log*, characterized by a sample time of 1 second, like in simulation; however, they are of general validity, in the sense that analogous results were obtained considering other data sets (eventually with small adjustments of the noise variances). Note that all filters were tuned according to the procedure described in the previous chapter.
The tests described in [1] were performed on a puller assembled with a dual camera system for radius measurement.
The results exposed in this chapter are obviously restricted to filtering, since only state estimation can be applied offline; the data series required for this purpose are the noisy radius measurement and the pull rate applied to the puller, which are certainly available from the data provided from St. Peters. On the other hand experimental testing of the LQR is possible only in the closed loop system and therefore online; for this project such tests were not performed, this would be the next step. For these reasons in the following only state estimation is considered, and therefore the LQR parameterization adopted for the tests presented in [1] is not discussed.
The state estimation results obtained in simulation and shown in the previous chapter are very satisfying; for all filter approaches - old UKF, improved UKF and EKF, besides the Particle Filter - it is possible to tune the noise variances in order to obtain a perfectly white output prediction er-

---

[1] for lack of time the Particle Filter was not applied; however the considerations made on the state vector pdf based on simulated data, together with the behaviour of the other filters, presented in the following, are sufficient to affirm that Gaussianity holds approximately for the experimental case too

ror, and thus optimal estimation performance. However, although the data generation model and the filter model used in simulation are not exactly equal, they are quite similar and therefore the considered case is almost ideal. Thus the performance of the filters has to be verified on experimental data, “generated“ by the puller itself, that is by the true process. The capillary model (the original model for the old UKF, the improved model for the improved UKF and EKF) is only an approximate description of this true process and therefore we expect the experimental performance to be somehow poorer.
Figures 8.1 and 8.2 report the original radius and height estimations performed on the puller during the test run of Siegert, together with the set points defined for the LQR (which are reported only for orientation purposes). Furthermore Figure 8.1 contains also the measured radius. Figure 8.3 reports the pull rate applied to the puller. As mentioned above, the considered data set was taken from the logfile *cplqr-20100311T******.log*; since this file does not contain the melt temperature estimation, its evolution is not shown. The parameterization of the (old) UKF used for this test is discussed in the next section, where also its estimation is reconstructed.

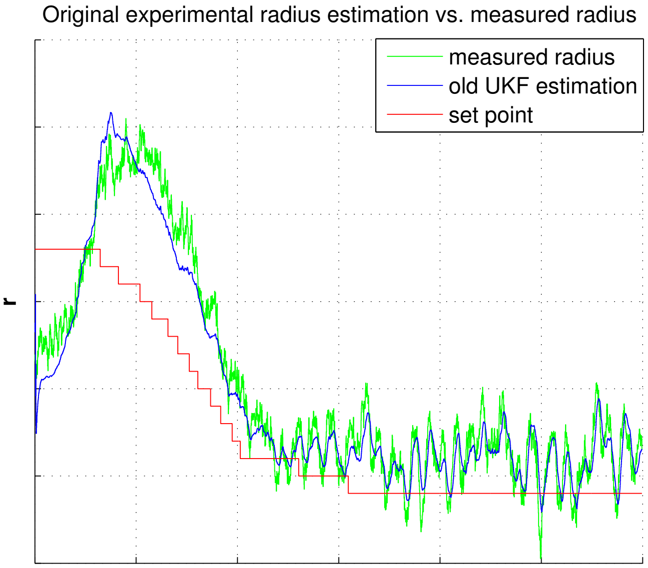

Figure 8.1: Original experimental radius estimation vs. measured radius

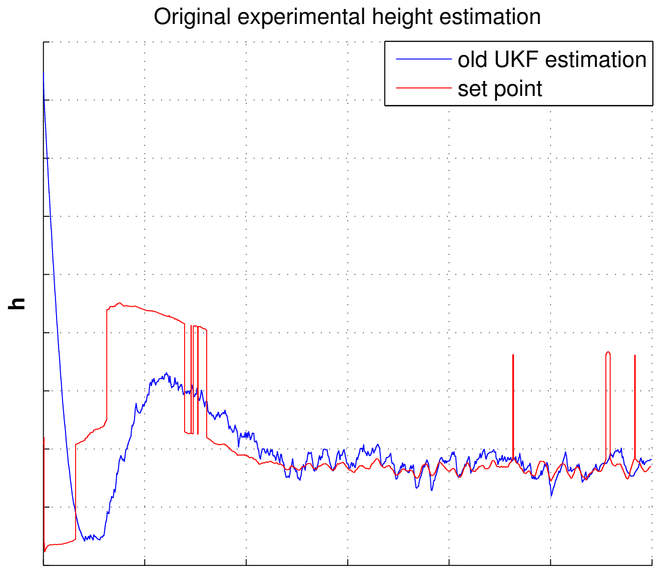

Figure 8.2: Original experimental height estimation

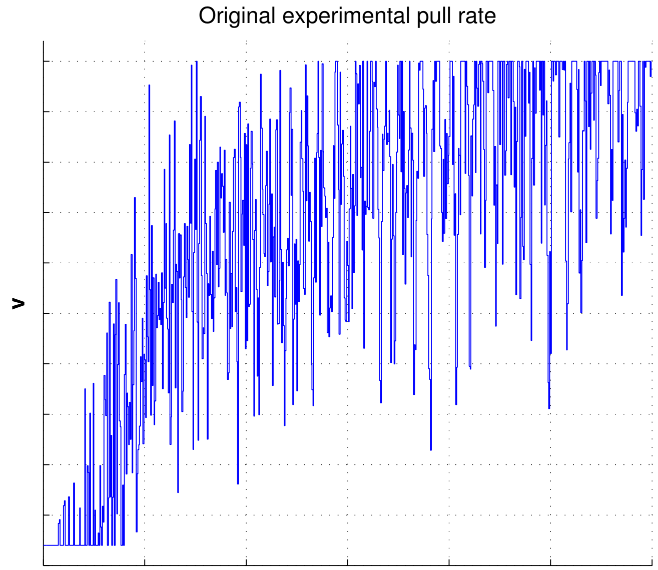

Figure 8.3: Original experimental pull rate

8.1 Standard tuning and retuning of old UKF

Initially the performance of the old UKF used for the closed loop test is considered. The following results refer to offline filtering, based on the measured radius and the pull rate of Figures 8.1 respectively 8.3. Three different noise variance sets are compared in the following, namely:

- Set 1
  The specific set of process and measurement noise variances adopted for the considered test run on the puller. This is the choice for which the estimations of Figures 8.1 and 8.2 were obtained (beside the initial conditions).

- Set 2
  The set of process and measurement noise variances declared as standard in [2], based on several simulation and experimental test runs.

- Optimal set
  The set deduced by the author retuning the old UKF according to the test for white noise (that is minimizing I_A). This is the optimal choice from the statistical point of view.

The results refer to the standard sigma point set parameters, that is $\alpha = 10^{-3}$, $k = 0$ and $\beta = 2$, and to the Tsivinskii formula for meniscus height approximation, since this is the original configuration adopted in [1]. Furthermore the nominal model parameter values were adopted in the filter model.

The initial estimation of the filter was chosen as follows ($P(0|0)$ was chosen appropriately, the exact numerical values are not important):

$$\hat{x}(0|0) = \begin{bmatrix} y(0) \\ h(0) \\ T_m(0) \end{bmatrix}.$$

Here $y(0)$ is the initial radius measurement and $h(0)$ was computed from this value with the Tsivinskii formula (for $\theta = \theta_C$). Furthermore, like for simulation, the output prediction error was considered only after the initial transitory (more precisely, the first 20 minutes are excluded for the white noise test) and therefore the impact of the initial conditions on the value of I_A is reduced. Anyway the problem of the initial conditions will be adressed more in detail in Section 8.3.

It is worth emphasizing once more that the noise variance choices of [2] were made considering not only the estimation, but also the LQR performance. Therefore it is not an optimal tuning from the point of view of state

estimation, but from the point of view of the behaviour of the closed loop system. Anyway, no statistical instruments were adopted for filter tuning and therefore it is highly empirical.
Figures 8.4, 8.6 and 8.7 report the comparison of the radius, meniscus height and melt temperature estimations for the three variance sets. Furthermore Figures 8.8 and 8.9 show the corresponding PSD and PS of the output prediction error $e(n)$. Finally Figure 8.5 reports the comparison between the measured radius and the radius estimation computed with the optimal set of noise variances.

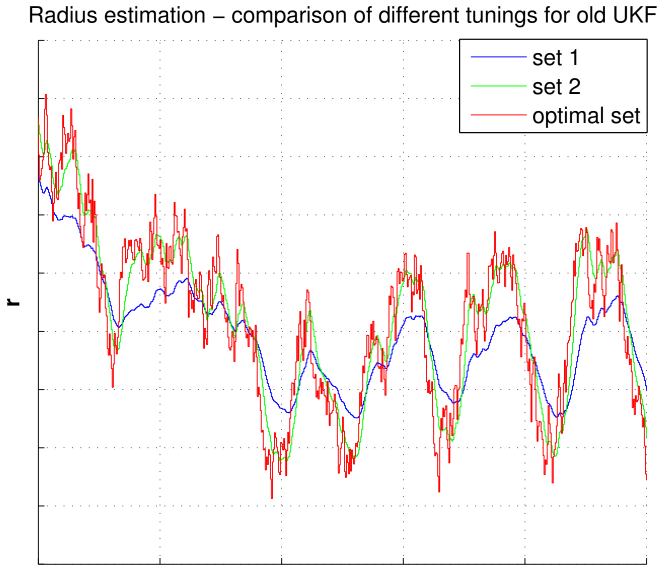

Figure 8.4: Radius estimation - comparison of different tunings for old UKF

From Figure 8.4 it turns out that the radius estimations are quite different. With the choices discussed in [2] (set 1 and set 2) the radius estimation is rather regular, that is the measured signal is filtered heavily (more precisely, it is a low pass filtering; the high frequency components of the radius measurement are eliminated). On the other hand, the best estimation from the statistical point of view (optimal set) yields a very irregular signal; as a matter of fact the radius measurement filtering is very weak. In fact for the sets 1 and 2 $R \gg q_{r/h}$, whereas for the optimal choice $R \ll q_{r/h}$. In the first case the high frequency variations/oscillations on the measured radius are attributed to the measurement noise, whereas in the latter case it is assumed that the oscillations are caused by the process itself; therefore they are not filtered out. This is confirmed by Figure 8.5, which compares

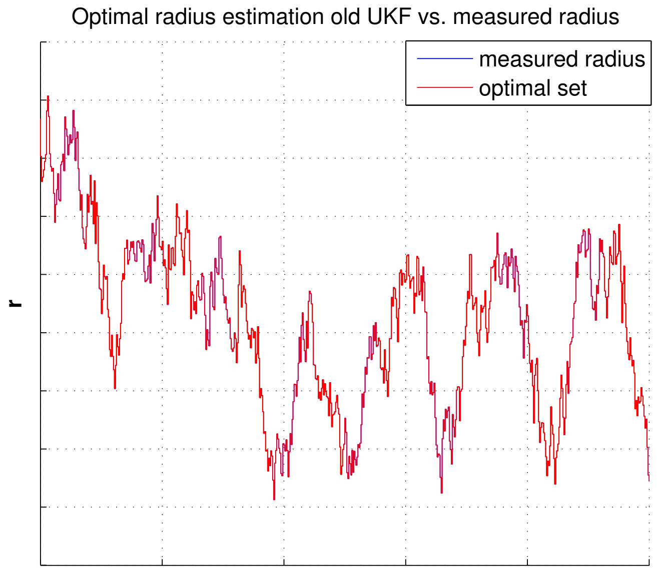

Figure 8.5: Optimal radius estimation old UKF vs. measured radius

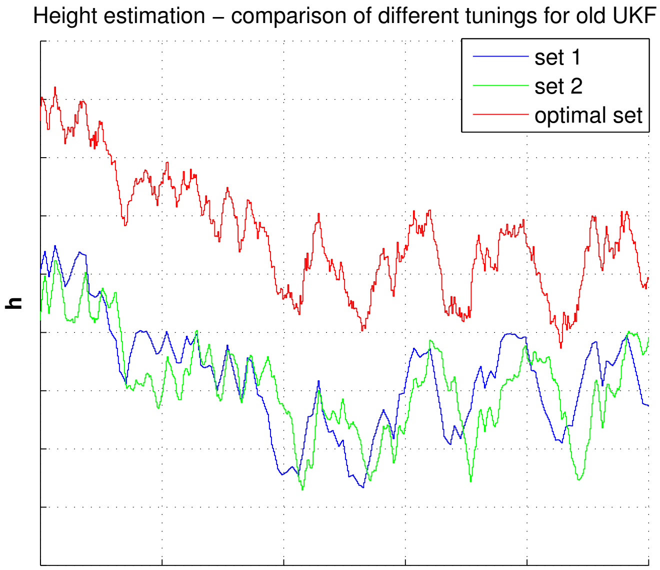

Figure 8.6: Height estimation - comparison of different tunings for old UKF

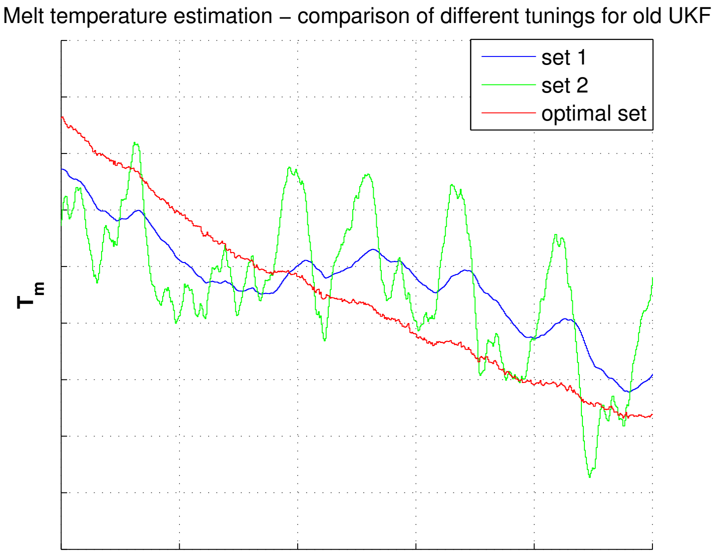

Figure 8.7: Melt temperature estimation - comparison of different tunings for old UKF

the optimal radius estimation to the measured radius; it turns out that they are identic! This is a quite surprising result; the best estimation quality is therefore obtained imposing a very weak (actually, a non-existent) filtering of the measured signal. Looking at Figures 8.8 and 8.9 there is no doubt that this is the optimal choice. The sets 1 and 2 yield a prediction error which is absolutely not white, and therefore from a statistical point of view they are not acceptable, whereas the optimal set prediction error is approximately white. Therefore for the state estimator it is "better" to consider the measurement very reliable; this means that probably the oscillations in the measured signal are caused by the process and not just by the measurement noise, as assumed by the filter tuning in [1]. However, as pointed out previously, it might be that this optimal choice is not applicable in the closed loop system since it could cause oscillations; therefore it has to be tested experimentally. The variance sets 1 and 2 yield quite poor statistical results, however they were tested in the experimental closed loop and therefore their compatibility with the LQR is already guaranteed.

Compared to the optimal tuning obtained in simulation, the output prediction error computed on experimental data is less white. As a matter of fact, in simulation for all filters it was possible to obtain $I_A < 1$, whereas in the experimental case the best choice yields $I_A \cong 20 \div 30$. Therefore on exper-

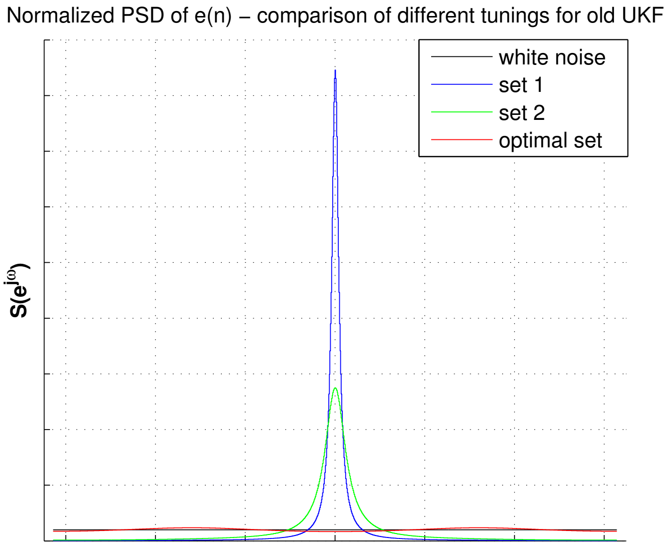

Figure 8.8: Normalized PSD of $e(n)$ - comparison of different tunings for old UKF

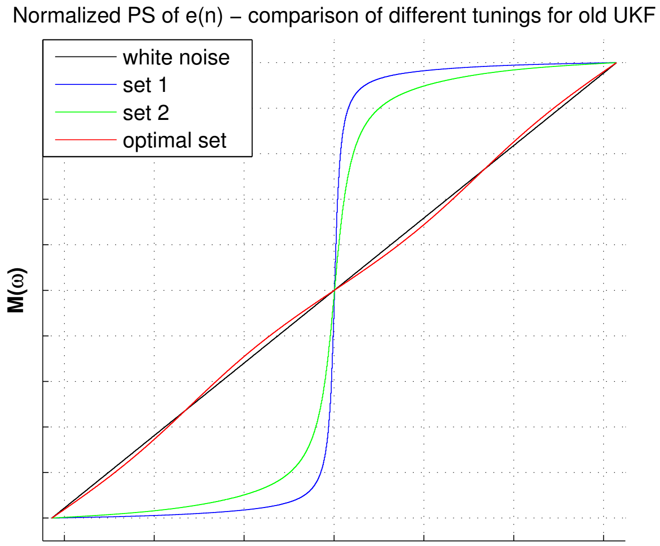

Figure 8.9: Normalized PS of $e(n)$ - comparison of different tunings for old UKF

imental data it is not possible to obtain perfectly white output prediction error; anyway, as can be seen from Figure 8.9, the optimal PS is quite close to the ideal one and therefore the whiteness condition is approximately satisfied. Actually, considering that the experimental data was generated by the puller and not by some kind of ideal model as for simulation purposes, it can be stated that the whiteness of $e(n)$ is more than satisfying.

The meniscus height and melt temperature estimations, reported in Figures 8.6 and 8.7 respectively, are rather different for the three noise variance choices. Since no information on the true evolution of these variables is available, looking just at the figures it is not possible to establish which estimation is the best one; but from the statistical white noise test we know that the most reliable, in terms of radius estimation, is the one given by the optimal set.

8.2 Comparison of optimal tunings for UKF and EKF

In the following the optimal tuning for the old UKF, described in the previous section, is compared to the optimal tunings for the improved UKF and the EKF; the values of I_A are, again, approximately the same for the different filters. All filters were initialized with the conditions reported above, beside the one on the third state variable: obviously for the improved UKF and the EKF this condition refers to the growth rate and no longer to the melt temperature. The initial growth rate estimation was computed, with formula (3.2), from the meniscus height and melt temperature conditions imposed for the old UKF. Finally for the model and sigma point set parameters the nominal values were considered, and the improved UKF and EKF rely on the Johansen-Boucher formula for meniscus height approximation.

Figures 8.10 - 8.13 report the comparisons of the radius and state variable estimations, whereas Figures 8.14 and 8.15 compare the normalized PSD and PS respectively.

Also for the improved UKF and the EKF the optimal tuning yields $R \ll q_{r/h}$; therefore also for these approaches the radius measurement is practically not filtered, as shown in Figure 8.10. As a matter of fact the *radius estimation* is identic for all approaches and indiscernible from the radius measurement; the graphs are perfectly overlapped. This does not hold for the meniscus height, growth rate and melt temperature estimations; these *inner state variable evolution estimations are very different from case to case, especially in the initial, transitory phase*. However, again, there is no possibility to establish which estimation of the hidden state variables is the best one; this can be figured out only on the actual puller, monitoring the closed loop behaviour.

Furthermore it is worth mentioning that the EKF approach exhibits an interesting common feature with the LQR; it relies on the same linearized model. It might be that for this reason its hidden state variables estimations are somehow more compatible with the LQR than the estimations computed by the UKF. However this is just a speculation, there is no guarantee that this is true; anyway this point is worth to be considered and eventually tested on the actual puller.

The discussed optimal tunings refer to a particular data set and therefore to a particular test run, characterized by specific pulling conditions, which change slightly from run to run (as a matter of fact, they are never exactly the same for two successive runs, even if the same puller and recipe is considered). However, as stated at the beginning of this chapter, the discussed tuning is of general validity. Several tests made on different data sets yielded that sometimes a small adjustment of the process noise variances, especially of the variance q_{T_m}/q_{v_g}, is necessary. However the differences are not dramatic, that is using the tuning presented in this section the output prediction error is always approximately white. Anyway it is illusory to search for the absolute minimum of I_A for each run, which is theoretically necessary since the pulling conditions are not perfectly constant over time. The choice discussed in this section, which is perfectly optimal for the considered run, is approximately optimal in general, and therefore it is a good starting point. If it should turn out that, contrary to expectations, the tuning is completely wrong, a small adjustment of q_{T_m}/q_{v_g} is sufficient to return to the optimal conditions, as experienced by considering different data sets. However, once the system is implemented on the puller, such adjustments should be necessary only periodically, as long as the process and the hardware configuration of the hot zone of the puller do not vary.

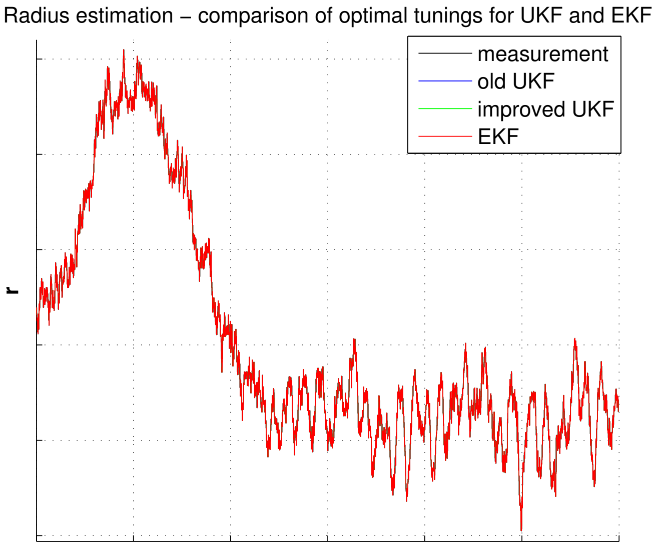

Figure 8.10: Radius estimation - comparison of optimal tunings for UKF and EKF

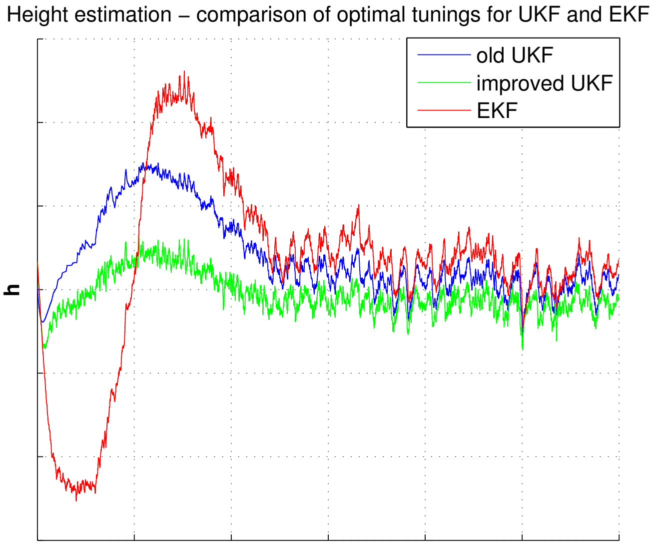

Figure 8.11: Height estimation - comparison of optimal tunings for UKF and EKF

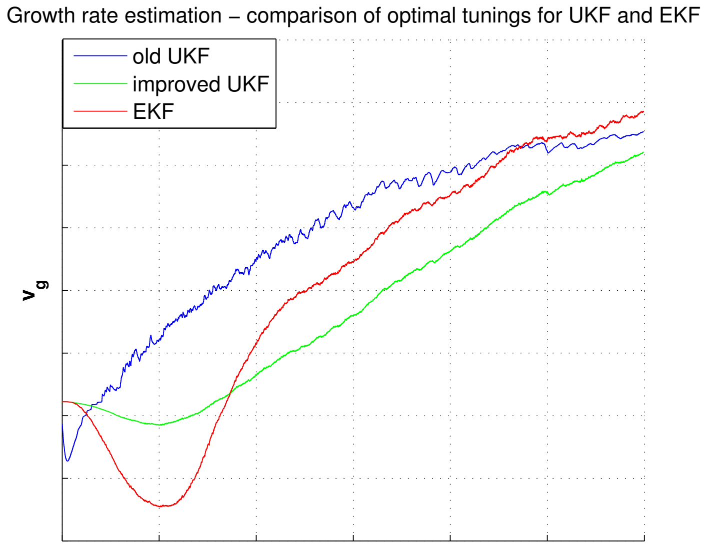

Figure 8.12: Growth rate estimation - comparison of optimal tunings for UKF and EKF

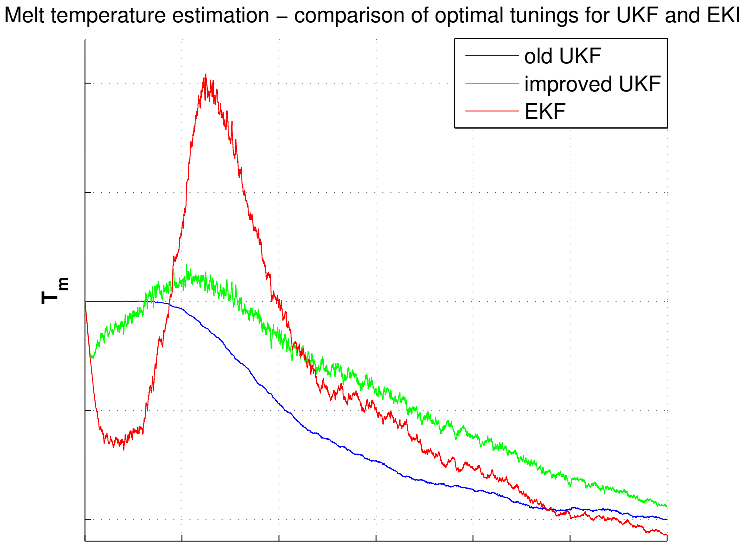

Figure 8.13: Melt temperature estimation - comparison of optimal tunings for UKF and EKF

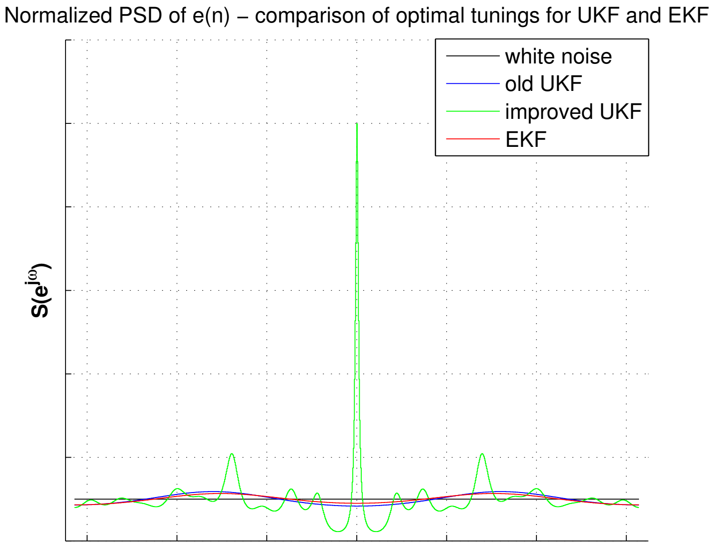

Figure 8.14: Normalized PSD of $e(n)$ - comparison of optimal tunings for UKF and EKF

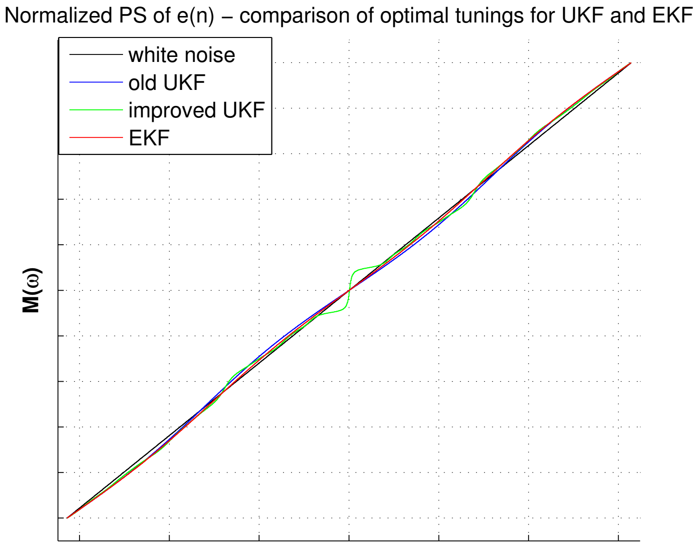

Figure 8.15: Normalized PS of $e(n)$ - comparison of optimal tunings for UKF and EKF

8.3 Filter initialization

The exposition of the state estimation results on experimental data is concluded with a brief discussion of the filter initialization, not yet considered in detail. Up to this point, limiting the attention to the improved UKF and the EKF, the state estimation was initialized as follows:[2]

- The radius estimation was initialized with the initial radius measurement, that is $y(0)$.
- The initial condition for the meniscus height estimation was computed with the Johansen-Boucher formula from $y(0)$, setting $\theta = \theta_C$.
- The initial condition for the growth rate was computed, with formula (3.2), from the initial conditions for h and T_m.

This initialization approach is justified by the following points:

- According to the tuning results, the radius measurement is quite reliable. Therefore it is rather logical to use it for radius estimation initialization.
- The meniscus height is a completely hidden variable and therefore we have no information about it. Thus a straightforward choice is to compute its initial value from the radius measurement, using the Johansen-Boucher formula. This formula requires the initial value of θ which is unknown too. The choice $\theta = \theta_C$ is a reasonable one.
- Also the growth rate is a hidden variable[3] and therefore computing the initial estimation with formula (3.2) is legitimate, since it is the only available information. The disadvantage of this equation is that it depends on all model parameters and not just on a and θ_C. Therefore the formula and thus the initial condition on the growth rate is affected by remarkable uncertainties.

The choice of the initial condition for the growth rate estimation is worth to be investigated more in detail. Formula (3.2) requires the initial value of the melt temperature T_m. On the puller, as long as state estimation starts when the seed is dipped into the melt, this value is (approximately) available from the measurement of the pyrometer. However, if the filter is (re)initialized at some point during the run, this approach is problematic,

---

[2] in the following the term "estimation initialization" is used for both the UKF and the EKF. However, to be exact, the initial conditions for the EKF refer to the prediction and not to the estimation, as pointed out in Section 4.5. Furthermore the considerations reported in this section hold for the old UKF too, considering T_m instead of v_g

[3] to be precise, it is approximately known since its evolution is similar to the pull rate evolution

since the temperature measurement is reliable only as long as the crystal is not in contact with the melt.[4]
Furthermore the melt temperature measurement is always affected by remarkable uncertainties, and therefore it has to be used very carefully. A straightforward approach to solve the problem associated with the temperature measurement is to use the current pull rate for the initialization of the growth rate estimation. We know that $v \neq v_g$, however usually the difference is not that large, and therefore as a first approximation this choice is admissible. With this approach the temperature measurement is no longer required.

Beside the choice of $\hat{x}(0|0)$ also the initial estimation error covariance $P(0|0)$ has to be considered.[5] Up to this point it was simply chosen as a diagonal matrix (denoted with the symbol P from now on), with appropriate numerical choices of the diagonal entries $p_r = p_h$ and p_{v_g}, since for the white noise test it is not a critical parameter. However this is clearly not the optimal choice from the point of view of the initial conditions, since in the discussion made above it was pointed out that the initial conditions on the meniscus height and the growth rate are affected by larger uncertainties than the one on the radius. Thus they should be described by a larger variance, whereas P assumes the same variance for the initial radius and meniscus height estimation (furthermore also the variance for v_g is too small with respect to the variance for r). Typically at the beginning such a simplified choice is made and the filter noise variances are tuned according to some statistical criterion which does not take into account the effect of the initialization (in our case the white noise test). Once the tuning is completed, the "steady-state" expression of the estimation error covariance $P(n|n)$, denoted in the following with P_{final}, is used for initialization, that is $P(0|0) = P_{final}$.[6] Although the EKF and the UKF tend to underestimate the estimation error covariance, this choice is normally by far more appropriate than standard choices like P. For instance, for the optimally tuned improved UKF P_{final} is given by a non-diagonal matrix:

---

[4] as a matter of fact, once the seed was dipped into the melt, the pyrometer points to the crystal and therefore it measures no longer the melt temperature

[5] for the EKF the initial conditions refer to $\hat{x}(0|-1)$ and $P(0|-1)$

[6] the term "steady-state" is not entirely correct since for nonlinear state estimation actually $P(n|n)$ becomes never completely constant. However after a certain time its variations become negligible in the case of interest, and therefore the steady-state condition is approximately achieved. Here P_{final} denotes the expression of $P(n|n)$ at the end of the estimation.

$$P_{final} = \begin{bmatrix} \bar{p}_r & * & * \\ * & \bar{p}_h & * \\ * & * & \bar{p}_{v_g} \end{bmatrix}.$$

The values of the diagonal entries are the most important ones. As a matter of fact these entries describe the scalar variances of the state estimation errors on r, h and v_g. The ratios between these variances differ significantly from the ratios described by the initial choice P. More precisely, the estimation error variances for h and v_g are increased significantly with respect to the variance for r, as expected from the uncertainty discussion made above. Furthermore the latter variance is given by the measurement noise variance R, another predictable result: we already know that basically the optimal radius estimation follows the radius measurement and therefore it is characterized by the same degree of uncertainty, that is by the variance of the measurement noise.

Figures 8.16, 8.17 and 8.18 show how the estimations of the improved UKF change if the estimation error covariance is initialized with P_{final} instead of P. No significant effect is obtained for the radius estimation (and neither for the radius prediction); furthermore, also I_A does not change significantly (it increases of an amount less than 1), as expected, since the initial 20 minutes of the trajectories are not used for the white noise test. On the other hand the transitory evolution of the height and growth rate estimation trajectories is deviated remarkably. After the transitory the estimations for $P(0|0) = P_{final}$ collapse with the estimations for $P(0|0) = P$; thus the "steady-state" phase, which is not affected by the initialization, begins.

These results confirm that filter initialization has to be considered apart from noise variance tuning. For the latter problem the white noise test on the output prediction error $e(n)$ is used, which does not take into account the transitory part of the output prediction error, that is the impact of the initial conditions is not considered. For the results exposed in this work only the portion of $e(n)$ with $n \geq 1200$ was considered, that is the first 20 minutes were excluded. However at least for the trajectories considered in this chapter the impact on I_A is not that dramatic since the radius estimation (and also the prediction) does not change significantly varying $P(0|0)$, as can be seen from Figure 8.16.

The considerations on $P(0|0)$, exposed for the improved UKF, are perfectly valid also for the old UKF and the EKF, with some small, obvious adjustments.

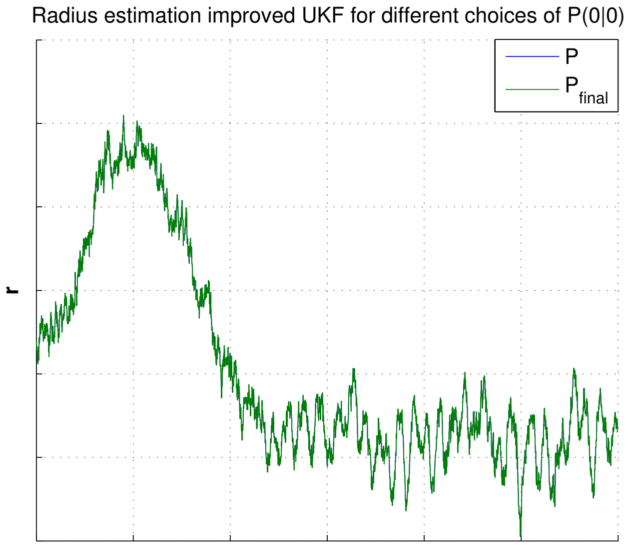

Figure 8.16: Radius estimation improved UKF for different choices of $P(0|0)$

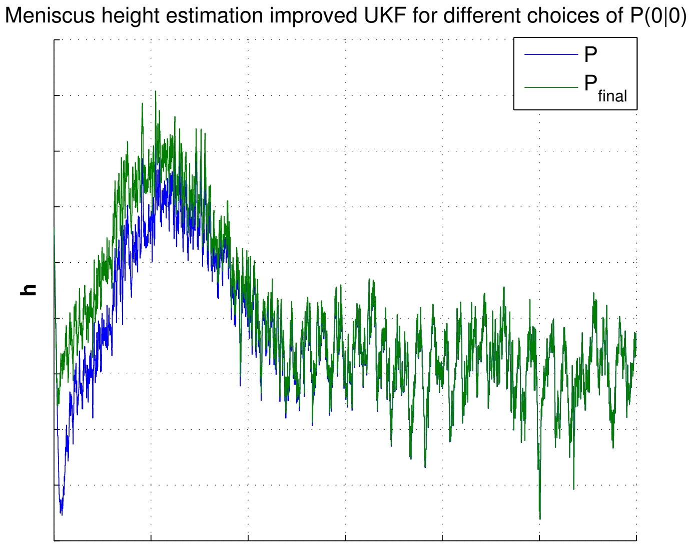

Figure 8.17: Meniscus height estimation improved UKF for different choices of $P(0|0)$

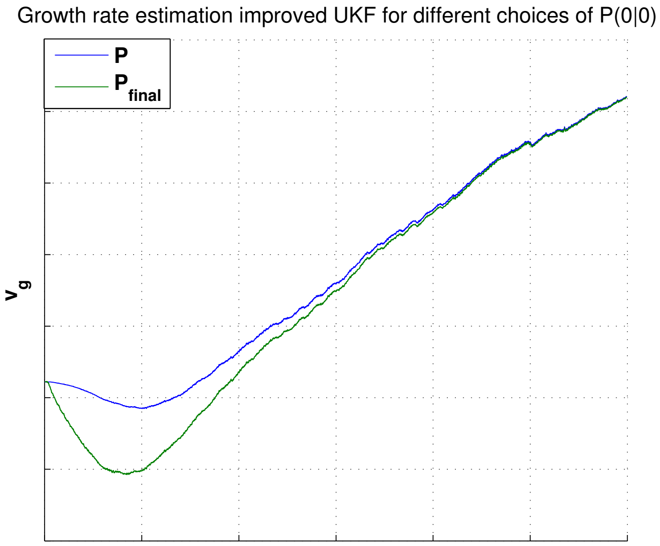

Figure 8.18: Growth rate estimation improved UKF for different choices of $P(0|0)$

Chapter 9

Conclusions and future work

The work exposed in this thesis is basically an extension and improvement of the model-based approach to radius regulation and state estimation developed by Siegert. In the first place a detailed theoretical analysis of his original capillary model, UKF and LQR design was performed; then the capillary model was corrected and refined, furthermore the LQR was simplified. The principal part of the thesis consisted in the detailed analysis of the nonlinear Bayesian state estimation problem. Besides the discussion of the original UKF used in [1] two more filter algorithms were exposed and explained, that is the EKF and the Particle Filter. These latter filters, as well as the new version of the UKF, are based on the improved capillary model. Finally a statistical method for filter performance evaluation was developed, namely the numerical area index based on the test for white noise on the output prediction error. This method allows a systematical tuning and comparison of the different filter approaches, on both simulated and experimental data.

As mentioned in the introduction, all the work done so far refers to the neck phase and was implemented in Matlab. By now it was not possible to test these improved methods on an actual puller.

Results and conclusions

The modifications applied to the capillary model led to a reduction of the nonlinearities and to an improvement of the robustness with respect to model parameter variations. As pointed out in Chapter 3 in the improved model only two parameters are left, namely the capillary constant a and the equilibrium growth angle θ_C, known with quite good approximation. Thus the dependence on highly uncertain parameters like the temperature gradient in the crystal was eliminated. This is a remarkable advantage, since both the LQR and the state estimator performance is highly dependent from the model accuracy. Thus the overall closed loop system robustness was improved.

Regarding the LQR some formal problems related with the original approach were solved, more precisely the intentional simplification which led to the usage of the input variable, the pull rate, as third state variable. More generally the usage of the improved capillary model led to a simplified, formally more correct, compact and computationally more efficient regulator. In simulation this approach works fine; however in order to understand what can be gained for real applications it has to be applied on the actual puller. Only there it will turn out how significant the benefits are.

The core of this thesis is the treatment of the nonlinear Bayesian estimation problem. The detailed analysis of the theoretical framework was one of the major concerns of the collaboration between MEMC Electronic Materials and the University of Padua, since it is indispensable for the understanding and handling of statistical state estimators like the UKF. Besides the importance of this analysis taken by itself, it generated some interesting insights. First of all a simpler filter algorithm, the EKF, was proved to yield the same performance as the UKF. Therefore a valid alternative to this latter estimator was found. Apart from the satisfying estimation performance the EKF relies on the same linearization procedure as the LQR and therefore it does not introduce any significant additional computation effort. Furthermore it could turn out that its hidden state variable estimations are somehow more compatible with the regulator than the ones of the UKF; this has to be verified on the actual puller.

Moreover the almost equal estimation performance of the old UKF,[1] the improved UKF and the EKF led to the conclusion that, despite the nonlinear dynamics of the capillary system, the conditions are almost ideal for state estimation, that is the state vector probability density function is almost Gaussian. This insight, formally proved with the application of the Particle Filter, should not be undervalued; as a matter of fact it demonstrates that for now investigating on more complex filter approaches or different sigma points sets for the UKF is not necessary, since in the conditions of interest nothing would be gained. Obviously for future modifications on the model this statement has to be revisited.

Another very important result of this thesis is the introduction of the area index for state estimation evaluation. Without such statistical evaluation methods the tuning and comparison of the filter approaches is very difficult and limited to empirical trial-and-error procedures. The simple index allows a systematic noise variance tuning; the optimal tuning is obtained by minimizing the index.

The analysis of the filter performances based on this index yielded some surprising results. As already mentioned the EKF and UKF performances

---

[1] however the old UKF is inferior from the point of view of the robustness with respect to model parameter variations

are almost the same. This is a quite unexpected result since the EKF relies on a rather simple linearization procedure; therefore generally the UKF works better. Another interesting result turned out applying the filter algorithms to experimental data from St. Peters: from a statistical point of view the best choice is to impose a very weak, almost non-existent filtering of the radius measurement! That is, for state estimation the best results are obtained considering the radius measurement y to be very reliable; thus the oscillations on y are probably caused by the process itself and not just by measurement noise. However this does not mean that the LQR has to deal with an unfiltered measurement signal. The closed loop system could (and will, most probably) work better with a suboptimal state estimation characterized by a stronger filtering of the radius measurement; alternatively the optimal noise variance configuration could be used, applying some kind of subsequent lowpass filtering to y. This has to be decided on the actual puller; furthermore only there it will turn out which filter algorithm computes the "best" estimation of the hidden state variables.

Suggestions for future work

Finally, based on the results exposed in this thesis, some suggestions for future work are given:

- The EKF and the improved UKF and LQR algorithms should be implemented and tested on an actual puller. All the results obtained so far have to be validated in the real closed loop system.
- Based on these experiments the filter and LQR tuning has to be revisited.
- The EKF and UKF performance should be compared in the real closed loop system too. It could turn out that the EKF is somehow more compatible with LQR control than UKF, since it relies on the same linearization procedure.

Ringraziamenti

Desidero ringraziare innanzitutto il mio relatore Prof. Beghi ed il mio correlatore Prof. Chiuso, per la grande disponibilità dimostratami durante l'elaborazione e la stesura della tesi.

Desidero inoltre ringraziare l'azienda MEMC Electronic Materials, Inc., per avermi dato la possibilità di svolgere il tirocinio. In particolare vorrei ringraziare il Per. Ind. Odorizzi, il Dr. Orschel ed il Dr. Siegert, ma anche tutti i ragazzi dello staff di ricerca e sviluppo che mi hanno dato una mano.

Inoltre ringrazio l'Ing. Peron per il suo sostegno nella fase iniziale del progetto.

Un sentito ringraziamento alla mia mamma, che, con il suo incrollabile sostegno morale ed economico, mi ha permesso di raggiungere questo traguardo; inoltre ringrazio mia sorella che ha dovuto sopportare i miei atteggiamenti spesso snervanti nei mesi passati.

Un affettuoso ringraziamento ad Angelika, che mi ha sostenuto nei momenti più difficili del lavoro e ha sempre creduto in me.

Infine un ultimo ringraziamento va ai miei amici e compagni di studio, in particolare a Klaus e Markus, per avermi accompagnato e aiutato durante il mio percorso di studio.

Bibliography

[1] M. Siegert. “LQR for Czochralski crystal pulling“. Internal report, MEMC Electronic Materials, St. Peters, Missouri, 2011.

[2] Personal communication with M. Siegert, MEMC Electronic Materials, St. Peters, Missouri, Oct. 2011 - Mar. 2012.

[3] J. Winkler, M. Neubert, J. Rudolph. “Nonlinear model-based control of the Czochralski process I: Motivation, modeling and feedback controller design“. *Journal of Crystal Growth*, vol. 312, issue 7, pp. 1005-1018, Mar. 2010.

[4] M. Neubert, J. Winkler. “Model based control of the Czochralski process“, presented at the 5^{th} Int. Workshop on Crystal Growth Technology, Berlin, Germany, Jun. 2011.

[5] Personal communication with V. Voronkov, MEMC Electronic Materials, Merano, Italy, Oct. 2011.

[6] G. Satunkin. “Modelling the dynamics and control design for Czochralski, Liquid Encapsulated Czochralski and Floating Zone processes“. *Progress in Crystal Growth and Characterization of Materials*, vol. 56, issues 1-2, pp. 1-121, Mar.-Jun. 2010.

[7] M.A. Gevelber, G. Stephanopoulos. “Dynamics and control of the Czochralski process: I. Modelling and dynamic characterization“. *Journal of Crystal Growth*, vol. 84, issue 4, pp. 647-668, Oct. 1987.

[8] T. Duffar (Editor). *Crystal Growth Processes Based on Capillarity: Czochralski, Floating Zone, Shaping and Crucible Techniques*. Chichester, United Kingdom: John Wiley & Sons Ltd, 2010.

[9] *Properties of Silicon. EMIS Datareviews Series No. 4*. London, United Kingdom: INSPEC, IEE, 1988.

[10] Fondazione Telios. “Glossario: Trasmittanza“. Internet: www.fondazionetelios.it/glossario/documents/Trasmittanza.pdf, Mar. 5, 2012 [Mar. 16, 2012].

[11] “The Physics Hypertextbook“. Internet: www.physics.info, [Mar. 16, 2012].

[12] T.H. Johansen. “Analysis of the crystal weighing method applied to Liquid Encapsulated Czochralski Growth“. *Journal of Crystal Growth*, vol. 84, issue 4, pp. 609-620, Oct. 1987.

[13] T.H. Johansen. “An improved analytical expression for the meniscus height in Czochralski growth“. *Journal of Crystal Growth*, vol. 141, issues 3-4, pp. 484-486, Aug. 1994.

[14] P.A. Laplante (Editor). *Comprehensive dictionary of electrical engineering*. Boca Raton, Florida: CRC Press, Taylor & Francis Group, 2005.

[15] W. Roonguthai, Wikipedia. “Silicon seed crystal puller rod“. Internet: en.wikipedia.org/wiki/File:Silicon_seed_crystal_puller_rod.jpg, Nov. 13, 2007 [Mar. 16, 2012].

[16] P. Schmid. “Scopriamo come viene creato un microprocessore“. Internet: www.tomshw.it/cpu.php?guide=20070425&page=la_nascita_di_un_processore-02, May 10, 2007 [Mar. 16, 2012].

[17] W.M. Higgins, G.W. Iseler, D.F. Bliss, G. Bryant, V. Tassev, I. Jafri, R.M. Ware, D.J. Carlson. “Improved phosphorus injection synthesis for bulk InP“. *Journal of Crystal Growth*, vol. 225, issues 2-4, pp. 225-230, May 2001.

[18] “Silicon crystal growing furnace - CG6000“. Internet: www.hellopro.co.uk/TPS_Groupe_ SPX_Corporation-4007-noprofil-2005057-20804-0-1-1-fr-societe.html, [Mar. 16, 2012].

[19] W. Lin, K.E. Benson. “The Science and Engineering of Large-Diameter Czochralski Silicon Crystal Growth“. *Annual Review of Materials Science*, vol. 17, pp. 273-298, Aug. 1987.

[20] W. Zulehner. “Historical overview of silicon crystal pulling development“. *Materials Science and Engineering: B*, vol. 73, no. 1, pp. 7-15, Apr. 2000.

[21] NDT Resource center. “Crystal defects“. Internet: www.ndt-ed.org/EducationResources/CommunityCollege/Materials/Structure/crystal_defects.htm, [Mar. 19, 2012].

[22] Personal communication with A. Giannattasio, MEMC Electronic Materials, Merano, Italy, Aug. 2011.

[23] G. Picci. *Filtraggio statistico (Wiener, Levinson, Kalman) e applicazioni*. Padua, Italy: Edizioni Libreria Progetto Padova, 2007.

[24] E.A. Wan, R. van der Merwe. “The Unscented Kalman Filter for Nonlinear Estimation“, in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, 2000, pp. 153-158.

[25] S.J. Julier, J.K. Uhlmann. “Unscented Filtering and Nonlinear Estimation“. *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401-422, Mar. 2004.

[26] L. Schenato. Class lecture, Topic: “Progettazione di sistemi di controllo: Lezione 1 - Ottobre 13, 2010“. Department of Information Engineering, University of Padua, Padua, Italy, Oct. 13, 2010.

[27] L. Schenato. Class lecture, Topic: “Progettazione di sistemi di controllo: Lezione 23 - Dicembre 7, 2010“. Summarized by R. Alberton, M. Ausserer, A. Barazzuol. Department of Information Engineering, University of Padua, Padua, Italy, Dec. 7, 2010.

[28] G. Picci. “Metodi statistici per l’identificazione di sistemi lineari“. Lecture notes of class “Identificazione dei modelli“, Department of Information Engineering, University of Padua, Padua, Italy, Jan. 2011.

[29] M. Sanjeev Arulampalam, S. Maskell, N. Gordon, T. Clapp. “A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking“. *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174-188, Feb. 2002.

[30] L. Schenato. Class lecture, Topic: “Progettazione di sistemi di controllo: Lezione 27 - Gennaio 12, 2011“. Summarized by C. Corfini, D. Fauri, E. Milesi. Department of Information Engineering, University of Padua, Padua, Italy, Jan. 12, 2011.

[31] J.D. Hol, T.B. Schön, F. Gustafsson. “On Resampling Algorithms For Particle Filters“, in *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE*, Sep. 2006.

[32] M. Steyvers. “Computational Statistics with Matlab“. Lecture notes of class “Computational Statistics with Matlab“, Department of Cognitive Sciences, University of California, Irvine, California, May 13, 2011.

[33] N. Whiteley. Lecture notes, Topic: “Sequential Importance Resampling“. Department of Mathematics, University of Bristol, Bristol, United Kingdom, Jan. 2012.

[34] L. Schenato. Class lecture, Topic: “Progettazione di sistemi di controllo: Lezione 26 - Gennaio 11, 2011“. Summarized by D. Volpato, G. Pattarello. Department of Information Engineering, University of Padua, Padua, Italy, Jan. 11, 2011.

[35] R. van der Merwe, E.A. Wan. "The Square-Root Unscented Kalman Filter For State And Parameter Estimation", in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, 2001 IEEE*, May 2001, pp. 3461-3464.

[36] L. Schenato. Class lecture, Topic: "Progettazione di sistemi di controllo: Lezione 24 - Dicembre 9, 2010". Summarized by F. Baro, A. Favero, C. Spitaler. Department of Information Engineering, University of Padua, Padua, Italy, Dec. 9, 2010.

[37] S. Pinzoni. "Note sui processi stocastici". Unpublished lecture notes of class "Stima e Filtraggio", Department of Information Engineering, University of Padua, Padua, Italy, Jan. 2010.

[38] F. Baro, A. Favero, C. Spitaler. "Controllo del comfort termoigrometrico in ambienti lavorativi moderati". Unpublished report for class "Progettazione di sistemi di controllo" of L. Schenato, Department of Information Engineering, University of Padua, Padua, Italy, Mar. 2011.

[39] "Crystal growth furnace". Internet: www.bridgat.com/crystal_growth _furnace-o265172.html, [Mar. 21, 2012].

[40] H. Tsuya: JPSJ Online-News and Comments. "Intrinsic Point Defects in Crystalline Silicon". Internet: jpsj.ipap.jp/news/jpsj-nc_17.html, April 10, 2006 [Mar. 20, 2012].

[41] Columbia University. "Silicon Basics - General Overview". Internet: https://www1.columbia.edu/sec/itc/ee/test2/silicon%20basics_files/ v3_document.htm, Sep. 11, 2001 [Mar. 20, 2012].

[42] Wikipedia. "Crucible". Internet: en.wikipedia.org/wiki/Crucible, Mar. 18, 2012, [Mar. 21, 2012].

[43] S. Meroli. "Two growth techniques for monocrystalline silicon: Czochralski vs Float Zone". Internet: meroli.web.cern.ch/meroli/Lecture_silicon_floatzone_ czochralski.html, [Mar. 21, 2012].

[44] SurfaceNet GmbH. "Czochralski (CZ) puller". Internet: www.surfacenet.de/html/czochralski__cz__puller.html, [Mar. 21, 2012].

[45] Schunk Group. "Crystal cultivation". Internet: www.engineersparadise.com/en/ipar/ 18423, [Mar. 21, 2012].

[46] Wikipedia. "Czochralski Process". Internet: en.wikipedia.org/wiki/File:Czochralski_Process.svg, Oct. 11, 2010 [Mar. 20, 2012].

[47] M.A. Gevelber, G. Stephanopoulos. “Dynamics and control of the Czochralski process: I. Modelling and dynamic characterization“. *Journal of Crystal Growth*, vol. 84, issue 4, pp. 647-668, Oct. 1987.

[48] P. Mutti, V. Voronkov. “Method for controlling growth of a silicon crystal to minimize growth rate and diameter deviations“. U.S. Patent 6 726 764 B2, Apr. 27, 2004.

[49] R. Irizarry-Rivera, W.D. Seider. “Model-predictive control of the Czochralski crystallization process. Part I. Conduction-dominated melt“. *Journal of Crystal Growth*, vol. 178, issue 4, pp. 593-611, Jul. 1997.

[50] R. Irizarry-Rivera, W.D. Seider. “Model-predictive control of the Czochralski crystallization process. Part II. Reduced-order convection model“. *Journal of Crystal Growth*, vol. 178, issue 4, pp. 612-633, Jul. 1997.

[51] Y. Cao. “Learning the Unscented Kalman Filter“. Internet: www.mathworks.com/matlabcentral/fileexchange/18217, Dec. 12, 2010 [Mar. 16, 2012].