UNIVERSITY OF CALIFORNIA SAN DIEGO

SAN DIEGO STATE UNIVERSITY

State Estimation for Control

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Engineering Sciences (Mechanical and Aerospace Engineering)

by

Mohammad S. Ramadan

Committee in charge:

University of California San Diego:

- Professor Robert R. Bitmead, Co-Chair
- Professor Nikolay Atanasov
- Professor Sylvia Herbert
- Professor David Sworder

San Diego State University:

- Professor Ke Huang, Co-Chair
- Professor Jerome Gilles

2023

The Dissertation of Mohammad S. Ramadan is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____

Co-Chair

_____

Co-Chair

University of California San Diego

San Diego State University

2023

DEDICATION

To the memory of my grandmother, Faheema. To her basils, mints, and olive trees. My hero as a child, I thought she was mine. My hero as an adult when I realized she was everyone's.

Time to rest, but you won't... I see you there,

Carrying Shadi, brushing little Tom's long hair.

EPIGRAPH

I’m not surprised . . .

-Nate Diaz

TABLE OF CONTENTS

LIST OF FIGURES

PREFACE

As an engineer by early construction, I had the understanding of how important estimation and filtering are. While it is in many situations fortunate that engineers can design well-behaved machines, such that PID controllers are efficient, the estimation problem seemed to me irreducible and less understood. Disturbances are mainly introduced by what we could not engineer or model, and therefore more obscure.

This dissertation is based on two questions asked by my advisor Professor Bitmead. The questions are "How can we use the Expectation Maximization algorithm in state estimation?" and "What is a good state estimate for control?", Chapters 2 and 3, respectively, are my answers. The first question made me dig deep into the statistics and Monte Carlo literature, while trying to maintain my control theory wording and notation. The latter was harder. For the second question, I realized, thanks to Charlotte Striebel [50], that correcting the wording is what I needed, the best A to B is an A such that B is best. Not accurate, but a telling analogy: my best partner is the one that makes me feel best. Therefore, the evaluation of control objective in selecting the state is necessary to answer the second question seriously. This was an invitation to stochastic control, the interaction between control and estimation, and understanding the difference between open- and closed-loop control.

ACKNOWLEDGEMENTS

If you look carefully enough, you can see beauty, and find heroes everywhere. I would like to thank many people for their existence in my life. First, my parents: Haifa and Saleh, thank you for your selflessness, your unlimited support and love.

Saying that I wouldn't be able to accomplish this without Prof. Bitmead is a fact. He made me a better reader, a better writer and a better thinker. Spending a weekly hour with him for several years was the catalyst of my PhD. Thank you Professor Bitmead!

I would also like to thank Prof. Ke Huang for his care, support, honest advice and passion towards helping me and others.

Thank you my committee members: Prof. David Sworder, for the support and kindness, Prof. Sylvia Herbert, for being genuinely nice and caring; Prof. Jerome Gilles, for the guidance and improving my math skills; Prof. Nikolay Atanasov, for his generous accurate detailed advice whenever I reached out.

It was great for me to be in a wonderful environment like MAE. Thank you to Prof. Mauricio de Oliveira and Prof. Raymond de Callafon for your great courses in control. I was lucky to teach with: Prof. Michael Davidson, and see closely his unmatched work ethics; Prof. Mike Tolley, his simplistic and honest life philosophy is something to keep in mind; Prof. Nicholas Boechler, who's not only a great instructor, but also a life coach; Prof. Stefan Llewellyn Smith, I'm better in fluid dynamics now; and Prof. Hieu Pham, whom I delivered my first lecture with. I would also like to thank Prof. Mamadou Diagne for his guidance and support of my post-PhD plans.

Thank you my great friends, my brothers, Mohammad Al Suwaidan and Ahmad Atallah for always being there for me, for all the running jokes we created, and all the laughs we had in the darkest days; Thank you Marshall Diaz, the other tiger with the less impressive muscle-ups; Mario Rodriguez and James Spradley, for sharing the same passion to grow; Daniel Maryanovsky, I really wish we knew each other earlier. Shruthi Nagabhushana and Sven Brüggemann, I appreciate that you listen and care. Mira Amr, Carlos Guzman, Anugrah Jo Joshy,

Zeaid Hasan, Peter Dorn, Kabir, Yi Zhao, Sam Crisafulli, Mohamed Shouman, and Bofu Zheng, thank you all for your support during my PhD journey ⸙

Chapter 2, in full, is a published material as it appears in: Mohammad S. Ramadan, Robert R. Bitmead "Maximum Likelihood recursive state estimation using the Expectation Maximization algorithm." Automatica, October 2022. The dissertation author was the primary investigator and author of this paper.

Chapter 3, in full, has been submitted for publication of the material as it may appear as: Mohammad S. Ramadan, Robert R. Bitmead, Ke Huang "State estimation for control: an approach for output-feedback stochastic MPC." The dissertation author was the primary investigator and author of this paper.

VITA

2016	Bachelor of Science, Jordan University of Science and Technology
2016-2017	Autopilot Design Engineer, Irbid, Jordan
2018–2023	Teaching Assistant, Department of Mechanical and Aerospace Engineering University of California San Diego and San Diego State University
2023	Doctor of Philosophy, University of California San Diego and San Diego State University

ABSTRACT OF THE DISSERTATION

State Estimation for Control

by

Mohammad S. Ramadan

Doctor of Philosophy in Engineering Sciences (Mechanical and Aerospace Engineering)

University of California San Diego, 2023
San Diego State University, 2023

Professor Robert Bitmead, Co-Chair
Professor Ke Huang, Co-Chair

Deterministic control theory is built on the presumptive luxury of complete access to the states. This premise is false in many applications. Hence, rendering questionable, the validity of the implemented control algorithms. On the other hand, acknowledging the lack or partial access to the states is an invitation to the realms of stochastic optimal control. A field of many platonic objects [38] and folk definitions, with hopes of advancement harshly met with computational intractability.

The dissertation focuses on finding suboptimal solutions to the stochastic control problem, via utilizing the machinery developed for deterministic control. This is done through reducing

the infinite dimensional information state, represented by the state filtered density, to one state value of (small) finite dimension. This reduction is guided by two arguments forming the bases of Chapters 2 and 3.

The first approach is built on statistical means, in which a point estimate, like the Maximum Likelihood Estimate, may have more claims to certainty equivalence in some applications than the typically used conditional mean. We derive a Maximum Likelihood recursive state estimator for non-linear state-space models, by combining the Expectation Maximization algorithm, and a particle filter. We prove that for nonlinear state-space systems with linear measurements and additive Gaussian noises, our formulation reduces to a gradient-free optimization in a form of a fixed-point iteration. The convergence properties of the sequences out of these iterations are inherited from the Expectation Maximization algorithm.

The second method engages directly with the control objective to achieve a state value or "estimate" that helps achieving this control objective. The State Selection Algorithm, which will be presented in Chapter 3, compiles the information about the state density, dynamics, constraints, and a given controller, and returns the best state value based on optimizing a prescribed finite-horizon performance function. The set of candidate states is provided by a particle filter. In the linear quadratic problem with polyhedral constraints, we show that the algorithm reduces to a quadratic program for the state value.

Chapter 1

Background

1.1 Introduction

As usually taught in undergraduate feedback control class, the dichotomy between open-loop and closed-loop control arises with the mere existence of any uncertainty [8]. Stability, robustness and disturbance rejection are the main attributes of feedback control. Since one rarely enjoys the luxury of perfect information and modeling of the dynamics and the incorporated exogenous disturbances, the necessity of these attributes becomes apparent.

Optimal feedback control laws are desirable, in large measure, due to their theoretically guaranteed stability and feasibility. Such attributes can be represented explicitly or implicitly in the cost function, or in the constraints, of the respective optimal control problem. The corresponding optimal feedback law is then guaranteed to have these attributes, by its construction.

A control designer may finish crafting their own cost function, then simply define the optimal control by writing $\arg\min$ to the left. Although this can be a valid mathematical definition of an optimal control law, whether it is physically realizable or computable, is a different story. One must remember that the mathematical well-definedness does not imply tractability, and in effect, many optimal control problems can end up as platonic objects that are impossible to solve for. Acquiring such control laws, that are valid over infinite horizons from any initial condition, requires solving a Dynamic Programming problem (tantamount to Hamilton-Jacobi-Bellman (HJB) partial differential equations in continuous time). This path has

proven early on to be prohibitively intractable for most applications of interest.

1.2 Deterministic optimal control

Many control design problems require, at first, the identification of a model, maybe a state-space model, from noisy data. We might be even required to design the experiment in which this data is acquired. Here we assume we are fortunate enough, and as control designers, the operators of the hidden technology [2], we are given a state-space model

$$x_{k+1} = f(x_k, u_k), \tag{1.1}$$

for which, u_k needs to be selected for any time-step k. The determination of u_k, if an optimal controller is sought, is guided by optimizing an objective function, typically of an additive form

$$J = \sum_{k=0}^{N} \ell_k(x_k, u_k), \tag{1.2}$$

where the ℓ_ks are non-negative stage costs. The control objective might also dictate satisfying input and state constraints, that is, $u_k \in \mathbb{U}$ and $x_k \in \mathbb{X}$, for all k. Notice that these constraints, and the value of J are in fact functions of the initial state x_0, and the control sequence $u_{0:N} = (u_0, \ldots, u_N)$. Therefore, the optimal control sequence $u^\star_{0:N}$ is simply defined as

$$u^\star_{0:N} = \arg\min_{u_{0:N}} J(x_0, u_{0:N}),$$

subject to

$$x_{k+1} = f(x_k, u_k),$$

$$x_k \in \mathbb{X},\ u_k \in \mathbb{U},\ k = 0, \ldots, N.$$

This control sequence is optimal if the initial state value is indeed x_0. For a different initial state, the above minimization problem has to be solved again for the corresponding optimal

control sequence. If we suppose that $\mathbb{U}$ is a finite set (probably a discretized version of an infinite set), with number of elements M, the number of possible open-loop control sequences is M^N. Even for a toy problem, with $M = 50$ and $N = 5$, there are more than 312 million different control sequences. Evaluating the corresponding cost of each trajectory, that is, finding the optimal control sequence by brute force, over the whole state-space, is impractical even for toy problems.

1.3 Dynamic Programming and the "curse of dimensionality"

Dynamic Programming (DP) is an algorithm (or can be seen as set of algorithms) to solve optimal control problems. DP does not reduce the searching space that we need to visit to find an optimal control sequence. However, from the data efficiency angle; repeated function evaluations are avoided through recording them. This is described as *Bellman's optimality principle*, which states that the tail sub-trajectories of an optimal trajectory are themselves optimal [52]. In other words, if the shortest path to a city B from city A passes through city C, the sub-path from C to B is the shortest starting from C. Therefore, if we know the optimal path from A to B, we saved ourselves finding the optimal path from any point on this $A \rightarrow B$ path to point B. DP utilizes the optimality principle, and the Markov property of the states. It propagates backwards in time, and finds solutions to the tail sub-problems first, which are then used to solve the original problem.

For the state-space system in (1.1) and the corresponding cost (1.2), the principle of optimality states that the tail sub-problems have to be optimal, hence, the last control u_{N-1} (typically the terminal stage cost is not a function of u_N, that is, with abuse of notation, $\ell_N(x_N, u_N) = \ell_N(x_N)$) must be optimal. The cost of the last step starting from different values of x_{N-1} in the state-space,

$$V_{N-1}(x_{N-1}) = \min_{u_{N-1}} \left\{ \ell_{N-1}(x_{N-1}, u_{N-1}) + \ell_N(x_N) \right\}.$$

This is the optimal cost (often denoted the cost-to-go) starting from any value of x_{N-1}. To define a recursion, we can let $V_N(x_{N-1}) = \ell_N(x_N)$, and therefore, the Dynamic Programming Equation at time-step $N-1$ is

$$V_{N-1}(x_{N-1}) = \min_{u_{N-1}} \left\{ \ell_{N-1}(x_{N-1}, u_{N-1}) + V_N(x_N) \right\},$$

and the Dynamic Programming Equation for any $k = 0, 1, \ldots, N-1$

$$V_{k-1}(x_{k-1}) = \min_{u_{k-1}} \left\{ \ell_{k-1}(x_{k-1}, u_{k-1}) + V_k(x_k) \right\}.$$

The causality condition dictates that any control, say at time k, must depend only on the information available up to time k. This is satisfied by Markov control laws, which has the form $u_k = u_k(x_k)$. Indeed, the optimal control $u_k^\star$, once the above Dynamic Programming recursion is completed, can be evaluated as in

$$u_{k-1} = \arg\min_{u_{k-1}} \left\{ \ell_{k-1}(x_{k-1}, u_{k-1}) + V_k(f(x_{k-1}, u_{k-1})) \right\},$$

and the right-hand-side is clearly a function of x_{k-1}.

The above Dynamic Programming recursion can be reduced into a difference Reccati Equation in linear Gaussian quadratic problems. However, in general, the state-space is to be discretized. If the state-space grid size is S, the Dynamic Programming algorithm is of complexity of $\mathcal{O}(NS^2M)$. This can be vastly reduced if the transition matrix (the dynamics restricted to the discretized grid) sparsity is exploited. The quadratic dependence on S, the size of the state-space grid and the fact that a grid of a size exponentially increasing with the state dimension is required for convergence, are what creates the so called "curse of dimensionality".

1.4 Open-loop control: No grids, no "curse of dimensionality"

The state-space grid is required in Dynamic Programming approach since the solutions of what we called the tail sub-problems are to be recorded along the way to solving the original problem. Differently from the optimality principle introduced by Bellman, Pontryagin introduced the maximum principle, which is restricted to deterministic optimal control problems and to finding open-loop solutions only.

Rather than blindly searching the input space, or gridding the state-space as in DP, the maximum principle presents the necessary and sufficient conditions for an optimal control. These conditions provide an alternative and less cumbersome way to find the optimal control. One can show, in the deterministic case, both principles yield the same control input sequence. The difference, however, starts to appear with the existence of state disturbances. Pontryagin's principle becomes inapplicable, while on the other hand, Dynamic Programming can be generalized to handle the stochastic, but full state-feedback case.

Although Pontryagin's principle and the developed toolboxes for deterministic optimal control and Model Predictve Control have shown great success, these toolboxes, in general, fail to handle stochastic systems, or nonlinear systems with nonconvex constraints. For such scenarios, the necessity of Dynamic Programming becomes apparent. DP as a grid based method, at least in principle, can handle nonlinearity and stochasticity (nonlinear dynamics replaced by discrete transitions and expectations, probabilities replaced by their discrete approximants over the grid).

The above discussion is another supporting example to a fact undergraduates learn in their control class; uncertainty requires feedback laws. Dynamic Programming is capable of handling uncertainties, since it provides feedback control policies. Furthermore, DP can naturally handle probabilities and expectations by simply approximating them by their discrete versions. On the other hand, open-loop control, which solves only for one trajectory and one control

sequence, fails to handle uncertainty, by its formulation.

Great effort has been made to solve the difficulties faced by the two approaches: the curse of dimensionality in the principle of optimality, and the inability to provide the feedback attributes: stability and robustness guarantees, when using the maximum principle. One idea relies on replacing the Dynamic Programming algorithm by stochastic approximation procedures [55], to form what is called Approximate Dynamic Programming [49], more famously Reinforcement Learning [42]. Another method is a very successful approach, conceptually simple and can handle complex systems and constraints, which is Model Predictive Control (MPC) [20]. In this approach, finite-horizon optimal control problem is solved in open-loop, online, that is, at each time-step, and the first control input in the optimal control sequence is applied.

1.5 Two great ideas behind MPC

Through solving an open-loop optimal control problem, at the current state only, MPC dodges the need to solve for the optimal control at many points in the state-space. The need for a function approximation procedure of a value function or an explicit feedback policy over the state-space [32] is therefore avoided. Furthermore, the open-loop solution offers more flexibility in handling constraints. From computational perspective, this set of arguments can be considered the first great idea of MPC. A less immediate, yet as important, idea of MPC is achieving closed-loop features through conducting open-loop solutions in a receding horizon fashion. Thanks to the receding horizon part, MPC yields asymptotic stability and recursive feasibility, under specific terminal conditions, over the infinite horizon. This is a foundational core result of Model Predictive Control due to Keerthi, and Gilbert [20], which is that properties of the open-loop finite-horizon can yield features of the closed-loop infinite horizon optimal feedback control.

It is a great success story, that of MPC. However, in the next sections, we provide a glimpse into problems, which are very realistic, but yet, cannot be handled by MPC.

1.6 Stochastic full-state-feedback control

In the previous sections, the dynamics considered, (1.1), are deterministic. Hence, open- and closed-loop control yield the same control sequences and state trajectories. Therefore, open-loop control can be used to avoid DP and its attendant, the curse of dimensionality. In this section however, state equation disturbance is considered, and the dynamics are no longer deterministic.

Suppose at time $k = 0$, we are given the following state space model

$$x_{k+1} = f(x_k, u_k, w_k), \tag{1.3}$$

where $x_k \in \mathbb{R}^{r_x}$ is the state, $u_k \in \mathbb{R}^{r_u}$ the control input and $w_k \in \mathbb{R}^{r_w}$ exogenous disturbance, with density $\mathcal{W}$. The control law is admissible if it satisfies the causality condition, and retains the Markovianness of the state equation (given that $\{w_k\}$ is white), that is, it has the form $u_k = u_k(x_k)$.

Proposition 1. *If the stochastic process $\{w_k\}$ is white, $\{x_k\}$ is Markov.* □

The requirement of having $\{x_k\}$ Markov stems from the fact that it is defined as the state, that is, the sufficient statistic in the following sense

$$p(x_{k+1} \mid x_0, \ldots, x_k, u_0, \ldots, u_k) = p(x_{k+1} \mid x_k, u_k),$$

in other words, out of the prior information as a whole, knowing x_k is sufficient to deduce the density of x_{k+1}, once u_k is chosen. This holds only if the process $\{w_k\}$ is white (since if it is not white, conditioning on w_k yields more information about x_{k+1}), and therefore we proceed with this assumption.

Suppose for the dynamics above, the states and inputs are to be within the constraint sets, $\mathbb{X}$ and $\mathbb{U}$, respectively, with $\epsilon \in [0, 1)$, an acceptable probabilistic constraint violation rate. If

the initial state value is x_0, MPC is implemented through first solving the following open-loop control problem

$$J(x_0) = \min_{u_{0:N-1}} \mathbb{E}_{w_{0:N-1}} \left\{ \sum_{k=0}^{N} \ell_k(x_k, u_k) \right\},$$
$$\textit{subject to } u_k \in \mathbb{U}, \text{“}x_k \in \mathbb{X}\text{”}, \textit{ for all } k = 0, \ldots, N,$$

where ℓ_ks are the stage costs, and the expectation is taken over all possible realizations of the process $w_{0:N-1} = (w_0, \ldots, w_{N-1})$. The constraint “$x_k \in \mathbb{X}$” needs a more careful treatment, since x_k as defined in (1.3) is random. In Chapter 3, we adopt the notion of chance constraints for constraints involving random state vectors.

We call the above problem a stochastic full state-feedback MPC problem. Although deterministic problems are the natural scope of MPC and its solvers, several engineering tricks can be made to stitch this gap. Typically, these tricks use a deterministic version of the problem, with information about the uncertainty used to tighten the constraints of the new deterministic problem, therefore replacing the stochastic MPC with a deterministic MPC with tighter constraints. These methods are called tube-based methods [24]. The formulation of tube-based methods is typically for linear systems with convex constraints. For general nonlinear systems, with general constraints, these methods can become inapplicable, as the constraint tightening procedures are complex and computationally demanding in such cases.

1.7 Partial state observation

Typically, the state is not directly accessible to measurement, and instead, we have access to some observation y_k. This observation conveys some information about the state x_k. In

mathematical terms, we have

$$x_{k+1} = f(x_k, u_k, w_k), \tag{1.4a}$$

$$y_k = g(x_k, v_k), \tag{1.4b}$$

where $x_k \in \mathbb{R}^{r_x}$ is the state, $u_k \in \mathbb{R}^{r_u}$ the control input and $w_k \in \mathbb{R}^{r_w}$, $v_k \in \mathbb{R}^{r_v}$ exogenous disturbances. The initial state x_0 has a density $p(x_0)$.

Notice that the state in (1.4a) is still Markov if $\{w_k\}$ is a white process, since the discussion after Proposition 1 is still valid. However, the prior information we considered before, that is, the set of past inputs and states, is not anymore so. Instead, prior information is represented by the set of inputs, observations and initial state density. Therefore it is a "Hidden" Markov process. We can only track the state densities instead of the true states.

It is typical to assume the stochastic processes, $\{v_k\}$ and $\{w_k\}$, each to be white, and that they are independent from each other and from x_0, the initial state. The whiteness of $\{w_k\}$ guarantees the Markovianness of $\{x_k\}$, per Proposition 1. The whiteness of $\{w_k\}$ and $\{v_k\}$ and their independence of x_0, guarantee the conditional independence of y_k, when conditioned on x_k.

Proposition 2. *For system* (1.4)*, if* $\{w_k\}$ *and* $\{v_k\}$ *are white, and* w_k, v_k *and* x_0 *are mutually independent for all* k*, then* y_k *conditioned on* x_k *is conditionally independent from both* $x_{0:k-1} = \{x_0, x_1, \ldots, x_{k-1}\}$ *and* $\mathcal{Y}_{0:k-1}$*, that is*

$$p(y_k|x_k, x_{0:k-1}, \mathcal{Y}_{0:k-1}) = p(y_k|x_k).$$

□

This is a special case of Lemma 1 in the next chapter. This result, together with the Markovianness of $\{x_k\}$, avoid unnecessary technical complications, especially in the derivation of the Bayesian filter.

The objective here is to find a control law $u_k = \kappa(\cdot)$ to minimize the finite-horizon

trajectory cost function J

$$J = \mathbb{E}_{x_0, w_{0:N-1}, v_{1:N}} \left\{ \sum_{k=0}^{N} \ell_k(x_k, u_k) \right\}, \tag{1.5}$$

subject to some suitable description of the constraints, where the expectation is with respect to a probability space of elements: x_0, w_k and v_k, for $k = 0, \ldots, N$. The control law u_k is admissible if restricted to the causal form $u_k = u_k(\mathcal{Z}_k)$, where $\mathcal{Z}_k = (y_0, \ldots, y_k, u_0, \ldots, u_{k-1})$ [5]. This restriction maintains the causality condition, that is, any decision must be dependent only on the information available then. Furthermore, this admits stochastic Dynamic Programming equation and the concept of information state, which will be discussed next.

1.8 Stochastic optimal control

For linear Gaussian systems with quadratic costs, the cost is explicitly a quadratic function of the state mean trajectory. An optimal stochastic controller under these settings is an LQR control of the deterministic state mean dynamics, and a Kalman filter tracking the state conditional density. This is the separation principle commonly attached with its famous example, LQG control [3]. This, however, does not hold in general, and the stochastic optimal control is a function of the information state [5], as will be shown in the following discussion.

1.8.1 Information state and the Bayesian filter

Given the the values of x_k and u_k,

$$p(x_{k+1} \mid x_k, x_{k-1}, \ldots, x_0, u_k, \ldots, u_0) = p(x_{k+1} \mid x_k, u_k),$$

which is a restatement of the fact of x_k being a state. However, the premise of knowing x_k is not true for general dynamics as in (1.4), where, typically, $y_k \neq x_k$. This leads to the concept of the *information state* [21], which at time-k, is the filtered density function $p(x_k \mid \mathcal{Z}_k)$.

The information state can be propagated through the Bayesian filter, which consists of two steps,

1. Time update:

$$p(x_{k+1} \mid u_k, \mathcal{Z}_k) = \int p(x_{k+1} \mid u_k, x_k) p(x_k \mid \mathcal{Z}_k)\, dx_k, \tag{1.6}$$

2. Measurement update

$$p(x_{k+1} \mid \mathcal{Z}_{k+1}) = \frac{p(y_{k+1} \mid x_{k+1}) p(x_{k+1} \mid u_k, \mathcal{Z}_k)}{\int p(y_{k+1} \mid x_{k+1}) p(x_{k+1} \mid u_k, \mathcal{Z}_k)\, dx_k}. \tag{1.7}$$

Notice that to move from the filtered density at time-k to $k+1$, the values of u_k and y_{k+1} are used. If, to simplify the notation, $\pi_{k|k} = p(x_k \mid \mathcal{Z}_k)$ and $\pi_{k+1|k} = p(x_{k+1} \mid u_k, \mathcal{Z}_k)$, these two steps combined define the mapping

$$\pi_{k+1|k+1} = T(\pi_{k|k}, u_k, y_{k+1}), \tag{1.8}$$

where T maps $\pi_{k|k}$ to $\pi_{k+1|k}$ using u_k in (1.6), then to $\pi_{k+1|k+1}$ using y_{k+1} in (1.7).

1.8.2 Stochastic Dynamic Programming

The stochastic optimal control consists of two distinct parts [13, ch. 25]:

- ❖ A Bayesian filter that tracks the state filtered density, which is the information state (also called the hyperstate [4]).
- ❖ A control law that produces u_k as a function of the state filtered density provided by the filter, and minimizes (1.5).

The causality condition dictates that the control input used is a function of the available information up to the time of deciding the value of this control input. This information is the

information state, or the state filtered density as we discussed above. The last decision to be made, u_{N-1}, has to minimize the last tail sub-problem, according to the principle of optimality. Therefore, the cost of the last step, given the information $\mathcal{Z}_k$ available then

$$\min_{u_{N-1}} \mathbb{E}\left\{\ell_{N-1}(x_{N-1}, u_{N-1}) + \ell_N(x_N) \mid u_{N-1}, \mathcal{Z}_{N-1}\right\}.$$

This expectation is taken with respect to all involved random variables, which are: x_{N-1}, w_{N-1} and v_N. The state equation disturbance w_{N-1} has a known density $\mathcal{W}$. Hence, given the filtered density of x_{N-1}, applying the time-update (1.6) yields the predicted density $p(x_N \mid u_{N-1}, \mathcal{Z}_{N-1})$. The measurement noise v_N is known as well, thus, the likelihood density $p(y_N \mid u_{N-1}, \mathcal{Z}_{N-1})$ can be determined, at least in principle. Therefore, the above term is solely a function of the filtered density, that is, the information state at $N-1$, $\pi_{N-1|N-1} = p(x_{N-1} \mid \mathcal{Z}_{N-1})$,

$$V_{N-1}(\pi_{N-1|N-1}) = \min_{u_{N-1}} \mathbb{E}\left\{\ell_{N-1}(x_{N-1}, u_{N-1}) + \ell_N(x_N) \mid u_{N-1}, \mathcal{Z}_{N-1}\right\}.$$

Writing the expectation in its integral form

$$V_{N-1}(\pi_{N-1|N-1}) = \min_{u_{N-1}} \int \left(\ell_{N-1}(x_{N-1}, u_{N-1}) + \ell_N(x_N)\right) \times$$
$$p(x_N, x_{N-1}, y_N \mid u_{N-1}, \mathcal{Z}_{N-1})\, dx_N dx_{N-1} dy_N.$$

Define $V_N(\pi_{N|N}) = \mathbb{E}\left\{\ell_N(x_N) \mid \mathcal{Z}_N\right\}$, then the last stage recursion

$$V_{N-1}(\pi_{N-1|N-1}) = \min_{u_{N-1}} \int \left(\ell_{N-1}(x_{N-1}, u_{N-1}) + V_N(T(\pi_{N-1|N-1}, u_{N-1}, y_N))\right) \times$$
$$p(x_N, x_{N-1}, y_N \mid u_{N-1}, \mathcal{Z}_{N-1})\, dx_N dx_{N-1} dy_N.$$

This recursion can be written for any $k = 0, 1, \ldots, N-1$, starting from the terminal function $V_N(\pi_{N|N})$ and solving backwards in time. It is called the stochastic Dynamic Programming

equation.

In the linear dynamics case, the information state can be equivalently described by the first two central moments, the state conditional mean and the covariance. If the random vectors x_0, w_k, v_k for all k, are Gaussian, the system is unconstrained and the stage costs are quadratic, V_k is indifferent to the state conditional covariance and is only a function of the state conditional mean. However, for general systems, stochastic Dynamic Programming equation is intractable, not because of the curse of dimensionality over the states, x_ks, but much worse, the curse of dimensionality over the information states, the $\pi_{k|k}$s.

As explained in [53], the capability of approximating the conditional density, does not solve half the problem. Although stochastic optimal control admits the separation of the filter and the control, the intractability is mainly due to the curse of dimensionality, since the information state in the case of a Bayesian filter is of infinite dimension. In the following chapters, we rely on a particle filter to provide us with the information state, as a set of particles or weighted particles. Although such representations of the information state are not of infinite dimensions, they typically consist of a large number of particles, and hence they still require reduction.

1.8.3 Particle information state

In this section, a basic particle filter algorithm is presented as a convenient approximant of the Bayesian filter. It is in effect the generator of the information state as will be used in the subsequent chapters.

Particle filtering is a sequential importance sampling/resampling method [13], which exploits the strong law of large numbers to evaluate integrals by propagating many state estimates, called particles.

The dynamics in (1.4) can be written in an equivalent form using transition and measurement densities [46] $p(x_{k+1} \mid x_k, u_k)$ and $p(y_k \mid x_k)$, respectively, deduced from the densities $\mathcal{V}$ and $\mathcal{W}$, and the dynamics f and g in (1.4), and starting from the initial state density p_0, which is given.

A basic particle filter algorithm consists of the same two major steps of the Bayesian filter,

1. The time update step, in which, the set of particles $\{x_{k|k,j}\}_{j=1}^{L}$ are propagated through the state equation (1.4a), with a chosen u_k, using L realizations of the disturbance $w_{k,j} \sim \mathcal{W}$ and the resulting particles are denoted $\{x_{k+1|k,j}\}_{j=1}^{L}$. That is, given the particle filtered density

$$p(x_k \mid \mathcal{Z}_k) = \sum_{j=1}^{L} \delta(x_k - x_{k|k,j}), \tag{1.9}$$

where δ is the delta function, the predicted density can be evaluated through propagating the particles $\{x_{k|k,j}\}_{j=1}^{L}$ via

$$x_{k+1|k,j} = f(x_{k|k,j}, u_k, w_{k,j}),$$

where $\{w_{k,j}\}_{j}^{L}$ are $L-$independent realizations of w_k.

2. The measurement update step, when y_{k+1} becomes available, consists of computing the importance weights $\{\Omega_{k+1,j}\}_{j=1}^{L}$ by:

$$\Omega_{k+1,j} = \frac{p(y_{k+1} \mid x_{k+1|k,j})}{\sum_{j=1}^{L} p(y_{k+1} \mid x_{k+1|k,j})},\ j = 1, \ldots, L. \tag{1.10}$$

This equation is a direct consequence of substituting the predicted density evaluated from the time-update step

$$p(x_{k+1} \mid u_k, \mathcal{Z}_k) = \sum_{j=1}^{L} \delta(x_{k+1} - x_{k+1|k,j}),$$

in the measurement update step of the Bayesian filter (1.7)

$$\begin{aligned} p(x_{k+1} \mid \mathcal{Z}_{k+1}) &= \frac{p(y_{k+1} \mid x_{k+1}) p(x_{k+1} \mid u_k, \mathcal{Z}_k)}{\int p(y_{k+1} \mid x_{k+1}) p(x_{k+1} \mid u_k, \mathcal{Z}_k)\, dx_k}, \\ &= \frac{p(y_{k+1} \mid x_{k+1}) \sum_{j=1}^{L} \delta(x_{k+1} - x_{k+1|k,j})}{\int p(y_{k+1} \mid x_{k+1}) \sum_{j=1}^{L} \delta(x_{k+1} - x_{k+1|k,j})\, dx_k}, \\ &= \frac{p(y_{k+1} \mid x_{k+1}) \sum_{j=1}^{L} \delta(x_{k+1} - x_{k+1|k,j})}{\sum_{j=1}^{L} p(y_{k+1} \mid x_{k+1|k,j})}. \end{aligned}$$

We want this filtered density, $p(x_{k+1} \mid \mathcal{Z}_{k+1})$, to be represented by the particle set $\{x_{k+1|k,j}\}_{j=1}^{L}$. We have

$$\begin{aligned} p(x_{k+1|k,i} \mid \mathcal{Z}_{k+1}) &= \sum_{j=1}^{L} \frac{p(y_{k+1} \mid x_{k+1|k,i}) \delta(x_{k+1} - x_{k+1|k,j})}{\sum_{j=1}^{L} p(y_{k+1} \mid x_{k+1|k,j})}, \\ &= \frac{p(y_{k+1} \mid x_{k+1|k,i})}{\sum_{j=1}^{L} p(y_{k+1} \mid x_{k+1|k,j})}, \end{aligned}$$

for $i = 1, \ldots, L$. This is gives the importance weights updating formula in (1.10).

The conditional mean at any time k can be found by evaluating the weighted sample average, which under regularity conditions,

$$\mathbb{E}\left[x_k \mid y_0, \ldots, y_k\right] \approx \sum_{j=1}^{L} \Omega_{k,j} x_{k|k-1,j}.$$

Notice that we started this particle filter algorithm, in the time-update, with a set of unweighted particles, or more accurately, with equal weights $1/L$ each. However, we ended the measurement update step with a filtered density represented by the particles $\{x_{k+1|k,j}\}_j$ and their weights $\{\Omega_{k,j}\}_j$. We assume a resampling step is done at each time-step, such that the filtered

density at time-step $k+1$ is represented by the equally weighted particles $\{x_{k+1|k+1,j}\}_j$

$$p(x_{k+1} \mid \mathcal{Z}_{k+1}) = \sum_{j=1}^{L} \Omega_{k,j}\delta(x_{k+1} - x_{k+1|k,j}), = \sum_{j=1}^{L} \delta(x_{k+1} - x_{k+1|k+1,j}).$$

The resampling step

The resampling step is achieved through generating the cumulative distribution function of the weighted particles $\{x_{k+1|k,j}\}_j$ and $\{\Omega_{k,j}\}_j$. To avoid sorting the particles, we generate a cumulative distribution over the index set $\{1, 2, \ldots, L\}$

$$F(i) = \sum_{j=1}^{i} \Omega_{k,j}.$$

Figure 1.1 shows a cumulative distribution function of 50 weighted particles, the weights of these particle were randomly generated then normalized such that they add up to 1.

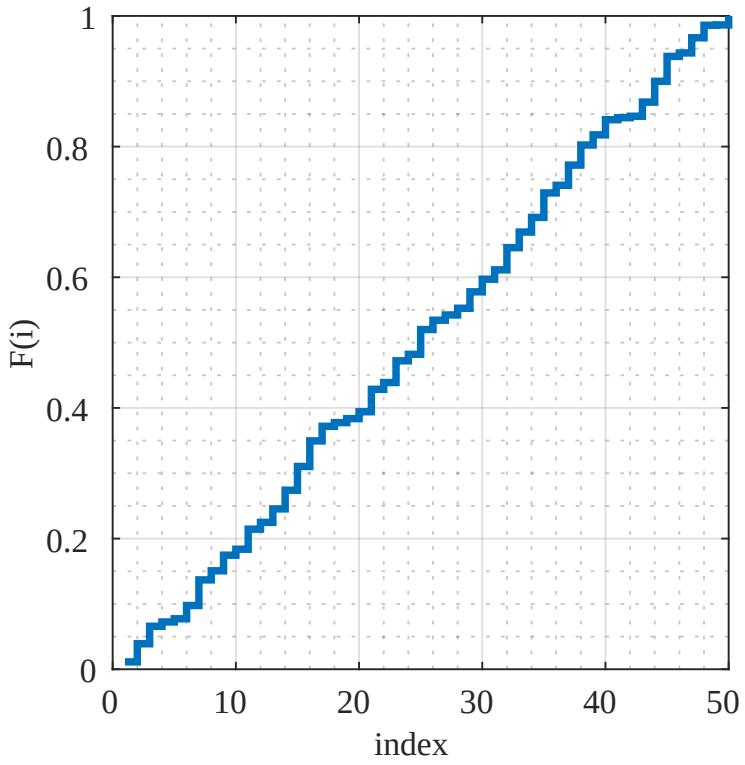

Figure 1.1. The cumulative distribution function of 50 particles, that is, $F(i)$ for $i = 1, \ldots, 50$

The resampling step proceeds by sampling a random variable, say r_j from the uniform distribution *uniform*$[0, 1]$. Then

$$x_{k+1|k+1,j} = x_{k+1|k,i}, \ \textit{if}\ r_j \in (F(i-1), F(i)],$$

and for $i = 1$ we define $F(-1) = 0$. Notice that the probability of picking a particle $x_{k+1|k,i}$ is the probability of having $r_j \in (F(i-1), F(i)]$, which is $F(i) - F(i-1) = \Omega_{k+1,i}$. Therefore, the probability of resampling a particle is this particle's weight.

1.9 Certainty Equivalence, a wormhole from deterministic to stochastic control

Certainty Equivalence (CE), similarly to the separation principle, suffers from the lack of a unique, agreed upon, universal definition. We therefore introduce our own folk definition, which we will be using in the next discussion and reasoning.

> Certainty Equivalence is the "action" of replacing the information state by a single state value, when solving for an optimal control law in (1.5).

If CE is adopted, the reduction of the information state to a single state value allows us to avoid stochastic Dynamic Programming and instead use the machinery developed for full-state feedback systems. Although this reduction is wrong, in general, and a deviation from the original dynamics (1.4), the resulting alleviation, of the intimidating computational and analytical complexities, is appealing for a control designer.

In our work, we use CE and enjoy the luxury of having a state value to feed to a full state feedback controller. However, this state value is selected with reasoning to help achieve a control objective. In Chapter 2, we reduce the information state (or the particle information state) to a Maximum Likelihood Estimate of the state. This estimate can serve as the state value to be used under CE, since in many applications, Maximum Likelihood state estimates may have more justifications under CE compared to other statistical point estimates, such as the conditional mean. For instance, the Maximum Likelihood Estimate is always in the support set of a density function, and might be better to handle heavy tails and skewness. In Chapter 3, we do not seek a statistical state estimate to use under CE, and instead, we select a state value that helps the closed-loop system, as a whole, achieve some control objective. This is done by formulating a cost function, not over control sequences, but over initial states of a fictitious closed-loop state sequence, which we will denote x'_k.

Chapter 2

Maximum Likelihood Recursive State Estimation using the Expectation Maximization Algorithm

Abstract

A Maximum Likelihood recursive state estimator is derived for non-linear state-space models. The estimator iteratively combines a particle filter to generate the predicted/filtered state densities and the Expectation Maximization algorithm to compute the maximum likelihood filtered state estimate. Algorithms for maximum likelihood state filtering, prediction and smoothing are derived. The convergence properties of these algorithms, which are inherited from the Expectation Maximization algorithm and the particle filter, are examined in two examples. For nonlinear state-space systems with linear measurements and additive Gaussian noises, it is shown that the filtering and prediction algorithms reduce to gradient-free optimization in a form of a fixed-point iteration. It is also shown that, with randomized reinitialization, which is feasible because of the simplicity of the algorithm, these methods are able to converge to the Maximum Likelihood Estimate (MLE) of multimodal, truncated and skewed densities, as well as those of disjoint support.

2.1 Introduction

We derive a maximum likelihood (ML) recursive state estimator for nonlinear systems based on: knowledge of the system equations and process noise and measurement noise densities; the operation of a particle filter to propagate predicted state conditional densities; and the Expectation Maximization (EM) algorithm [12, 37] to generate the ML filtered state estimate. If the state at time k is denoted x_k, and given the sequence of outputs up to time k, namely $\mathcal{Y}_{0:k} = \{y_0, y_1, \ldots, y_k\}$, the algorithm provides an estimate of the maximizer of the posterior state density $p(x_k|\mathcal{Y}_{0:k})$. This contrasts with Least-Squares estimators, which produce the conditional mean of this density. The contribution of this chapter is to provide an easily implementable algorithm which is recursive and which yields this Maximum Likelihood Estimate (MLE). We propagate the conditional densities, both prior and posterior, using a particle filter, and the EM algorithm is used to compute the MLE of the posterior density from the particles of the prior density, the current measurement and the known noise densities[1]. The observation is that knowledge of the noise densities permits computation of successive functions whose maximizers are non-decreasing in their likelihood with respect to the posterior density, which follows from the desirable convergence properties of the EM algorithm. If the system under consideration has additive Gaussian noises and linear measurement equation, the algorithm simplifies to a fixed-point iteration. This chapter clarifies these details in relation to the particle filter, the EM algorithm and MLE.

In many scenarios, the conditional state density can: be multimodal as when tracking a group of objects [36, 48]; have skewness, limited support, one-sidedness as in some Stochastic Volatility Models [54, 27]; or be governed by inequality state constraints [26], leading to truncated or disjoint support such as might arise in stochastic Model Predictive Control. The justification for the choice of the conditional mean as the point estimate of a filter may be weakened or render

[1]The extension to the case including a known exogenous input or control signal as part of the system equations is direct in a particle filter. Therefore, it is omitted here.

an estimate which is infeasible under the density. In such scenarios, the MLE may have more reasonable grounds for use. Since finding the MLE requires optimization over the posterior density, the recursive procedure involves two stages: the propagation of the state densities, for which we rely on a particle filter, and an iterative optimization to find the maximizer. However, the density provided by the particle filter is a collection of point masses, usually with the same singleton mass, and so is not suited to Newton-like optimization. A novelty of this chapter and its algorithms is that the previous time's posterior density plus the current measurement are combined with the known continuous densities of the process and measurement noises to yield a gradient based likelihood maximizer.

Recursive propagation of prior/predicted and posterior/filtered conditional state densities in nonlinear Bayesian filtering involves two distinct steps.

1. From the measurement equation, compute $p(y_k|x_k)$ and then, using the prior density, $p(x_k|\mathcal{Y}_{0:k-1})$, compute the weighted intermediate product $p(y_k|x_k)p(x_k|\mathcal{Y}_{0:k-1})$, which is then normalized to create the posterior density. Bayes tells us that

$$p(x_k|\mathcal{Y}_{0:k}) = \frac{p(y_k|x_k)p(x_k|\mathcal{Y}_{0:k-1})}{\int p(y_k|x_k)p(x_k|\mathcal{Y}_{0:k-1})\,dx_k}.$$

2. Using the state equation and the process noise density, the posterior (filtered) density, $p(x_k|\mathcal{Y}_{0:k})$, is propagated to the next prior density, $p(x_{k+1}|\mathcal{Y}_{0:k})$.

For the bootstrap particle filter, the measurement-update or filtering Step 1 is achieved by resampling the original particles under the product density. The time-update step involves propagating each particle through the state equation and sampling from the process noise density. In our algorithm, we apply EM directly to maximize the intermediate product to produce the filtered MLE.

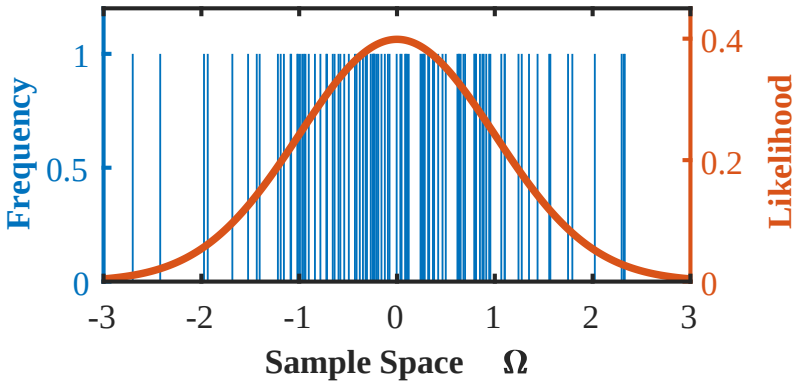

Figure 2.1. In blue is the histogram of 100 particles independently sampled according to a standard normal, $\mathcal{N}(0,1)$, the red curve.

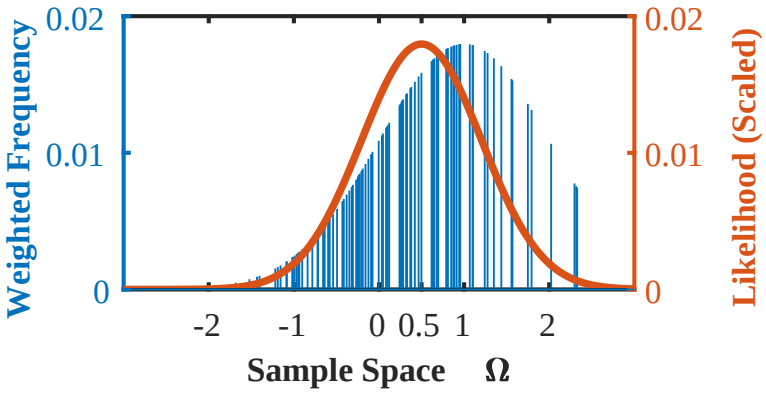

Figure 2.2. The same 100-particle distribution weighted by a Gaussian of mean 1 and variance 1, $\mathcal{N}(1,1)$. The red curve is the pointwise product $\mathcal{N}(0,1). \times \mathcal{N}(1,1)$, whose maximizer is sought and is the (unknown) unnormalized likelihood describing the weighted particles.

Figure 2.1 is a histogram in blue which depicts 100 particles (samples) drawn from a standard normal density, $\mathcal{N}(0,1)$, which is in red. The spatial distribution of the particles captures the underlying density. Since the density is continuous and the histogram has infinitesimally wide bins, each particle has frequency one. So, the particle frequency cannot directly be used to determine the likelihood maximizer. Similarly, Figure 2.2 shows the intermediate product of densities, where each particle in Figure 2.1 is multiplied by its corresponding value of $p(y_k|x_k)$, here $\mathcal{N}(1,1)$. The red curve is the (unknown) product function, the unnormalized filtered density, in this example also of Gaussian shape. The aim of an ML state estimator is to determine the

maximizer of the red curve from these weighted blue values. For us, this involves successive smoothing over the weighted particles in the E-step followed by maximization in the M-step.

In [31], a piecewise continuous cumulative distribution function is constructed from the weighted particles approximating the filtered state density $p(x_k|\mathcal{Y}_{0:k})$, limited to the case when the state-space is one-dimensional. In [11], this solution is extended to a multi-dimensional state-space. However, it has a computational complexity of $\mathcal{O}(N^2)$, where N is the number of particles, and a nonstandard particle filtering scheme, as reported in [18]. Earlier, in the construction of the auxiliary particle filter [39], it was shown that it is possible to construct a smooth approximation of $p(x_k|\mathcal{Y}_{0:k})$ under the name of "the empirical filtered density" using the weighted particles approximating $p(x_{k-1}|\mathcal{Y}_{0:k-1})$ and the current measurement. We extend this construction with the computation of an approximate smooth function, Q, as shall be explained shortly. This constructed smooth function could be directly maximized using various search methods and the new calculation permits gradient search for the MLE. We introduce the EM algorithm as an iterative procedure to achieve this, perhaps with multiple initial starting values in the multimodal case. This yields a sequence of estimates with non-decreasing likelihood which converges to a stationary point [57, 12]. This will be supported by computational examples in Sections 2.6.1 and 2.6.2.

In effect, the EM algorithm smooths and recenters the empirical filtered density in the E-step, taking into account the weights of the particles and their spatial density, before maximizing at the M-step. The iterations refine and localize the smoothing to approach the MLE. Ultimately, this is limited by the quality of the empirical density.

The EM algorithm [12] is versatile with many variations [34, 56, 23] and applications primarily in parameter estimation. It consists of two major steps: the expectation (E) step, in which an approximant to the log-likelihood function is constructed, followed by the maximization (M) step, which seeks to find the maximizer of this function. Hence, EM iterations deliver a sequence of functions for which their maximizers are non-decreasing with respect to the likelihood function under consideration [57].

Contribution

Equipped with Monte-Carlo particle methods, the EM algorithm is widely used in parameter estimation for nonlinear non-Gaussian systems [46, 40, 28, 25]. We examine the use of EM in the context of state estimation; to our knowledge this is a novel application and different in principle, since, unlike parameters, the components of the state are dynamically changing according to the tailored system model. Because of this, the MLE estimator needs to be recursive and to forget or deemphasize data from the distant past. This is achieved by the particle filter under suitable mixing conditions. For parameter estimation, prediction, filtering and smoothing are indistiguishable; not so for state estimation. Here is how our algorithm compares to the existing literature

- In [46], EM is used together with a particle smoother to evaluate the MLE of a constant parameter in a nonlinear state-space model, using a sequence of measurements.
- The work in [51] applies a recursive stochastic gradient search using a particle approximation of the gradient of the log likelihood function of the data with respect to a constant parameter.
- Differently from [46], our algorithm processes measurements recursively in time. And, differently from both [46] and [51], it is applied in state estimation which can be time-varying and stochastic, as opposed to constant parameters.

Outline

In Section 2.2, the underlying assumptions on the state-space model under consideration are elucidated. In Section 2.3, the EM state filter (EMSF) algorithm is introduced. It is shown, in Section 4, that EMSF reduces to a convergent fixed-point iteration method under mild assumptions. The extensions of EMSF to the contexts of state prediction and smoothing are provided in Subsections 2.5.1 and 2.5.2, respectively. In Section 2.6, EMSF is applied to two

examples: firstly, to a linear Gaussian problem where the Kalman filter also yields the MLE; and then to a nonlinear problem with multimodal state densities.

2.2 Formulation

Consider the following nonlinear dynamical system additive in the noises[2]:

$$x_{k+1} = f(x_k) + w_k,\ x_k \in \mathbb{R}^{r_x}, \tag{2.1a}$$

$$y_k = g(x_k) + v_k,\ y_k \in \mathbb{R}^{r_y}, \tag{2.1b}$$

$$w_k \sim W_k(.),\ v_k \sim V_k(.),$$

$$x_0 \sim p_0(.),$$

where,

- the integers $r_x > 0$ and $r_y > 0$ are the dimensions of the state and output vectors, respectively,
- the stochastic processes $\{w_k\}$ and $\{v_k\}$ are white and mutually independent, and their probability density functions, $W_k(\cdot)$ and $V_k(\cdot)$, are known and continuously differentiable for all k,
- x_k is the state vector, with initial value, x_0, having a known initial distribution which is independent of $\{w_k\}$ and $\{v_k\}$,
- y_k is the measurement vector,
- the functions f and g are known continuously differentiable functions.

[2]The notation $\sim$ here means "is distributed as the density on the right".

Problem Statement

Given the observation sequence, $\mathcal{Y}_{0:k} = \{y_0, y_1, \ldots, y_k\}$, the problem is to find the MLE of the state-vector x_k using the log-likelihood function $\log p(x_k|\mathcal{Y}_{0:m})$, for $m = k$ (filtering), $m > k$ (smoothing), and $m < k$ (prediction).

2.3 ML state filtering by the expectation maximization algorithm

For the problem statement at time k, given:

- the measurement y_k,
- the filtered density at the previous time $k-1$, i.e. $p(x_{k-1}|\mathcal{Y}_{0:k-1})$,
- initial time-k state estimate x_k^0,

we set iteration counter $i = 0$ and seek to find x_k^{i+1} such that $\log p(x_k^{i+1}|\mathcal{Y}_{0:k}) \geq \log p(x_k^i|\mathcal{Y}_{0:k})$ [57]. This is achieved by two major steps: the expectation step (E-step), followed by the maximization step (M-step). We continue with these iterations in i until we satisfy a convergence criterion, which is guaranteed by the non-decreasing property. At this stage, we either restart with a different initial condition, x_k^0, or move to time $k+1$.

2.3.1 E-step

The E-step is performed by evaluating the following expectation,

$$\begin{aligned} Q_x(x_k, x_k^i) &= \mathbb{E}\left\{\log p(x_k|x_{k-1}, \mathcal{Y}_{0:k})|x_k^i, \mathcal{Y}_{0:k}\right\}, \\ &= \int \log p(x_k|x_{k-1}, \mathcal{Y}_{0:k}) p(x_{k-1}|x_k^i, \mathcal{Y}_{0:k})\, dx_{k-1}, \end{aligned} \tag{2.2}$$

whose calculation in terms of available densities will be presented shortly in the next theorem.

Before presenting the theorem, we clarify a notational aspect: the function $p(x_k \mid x_{k-1}^i, \mathcal{Y}_{0:k})$, for example, is the probability density function of x_k conditioned on both x_{k-1}^i and

$\mathcal{Y}_{0:k}$. Also, note that

$$p(x_k, x_{k-1}^i|\mathcal{Y}_{0:k}) = p(x_k|x_{k-1}^i, \mathcal{Y}_{0:k})p(x_{k-1}^i|\mathcal{Y}_{0:k}).$$

After taking the logarithm of both sides,

$$\log p(x_k, x_{k-1}^i|\mathcal{Y}_{0:k}) = \log p(x_k|x_{k-1}^i, \mathcal{Y}_{0:k}) + \log p(x_{k-1}^i|\mathcal{Y}_{0:k}),$$

the second term in the right-hand-side is independent from x_k. Thus, maximizing either the the joint ($p(x_k, x_{k-1}^i|\mathcal{Y}_{0:k})$) or the conditional ($p(x_k|x_{k-1}^i, \mathcal{Y}_{0:k})$) likelihood functions leads to the same result in the M-step; since the maximization is executed with respect to x_k.

Lemma 1. *For system* (3.1) *at time $n \geq k$, y_n conditioned on x_k is conditionally independent from both $x_{0:k-1} = \{x_0, x_1, \ldots, x_{k-1}\}$ and $\mathcal{Y}_{0:k-1}$, that is*

$$p(y_n|x_k, x_{0:k-1}, \mathcal{Y}_{0:k-1}) = p(y_n|x_k).$$

Proof. From (3.1) and for $n \geq k$, y_n is a function of x_k, $w_{k:n-1}$ and v_n. Since $w_{k:n-1}$ and v_n are independent from $x_{0:k-1}$ and $\mathcal{Y}_{0:k-1}$, we have the result. □

This is a restatement of the Markovianness of the state equation (3.1).

Theorem 1. *Given the dynamical system (3.1), the filtered density $p(x_{k-1}|\mathcal{Y}_{0:k-1})$, the current measurement y_k, and x_k^i, the integral in (2.2) is, apart from a positive multiplicative constant and an additive constant both independent from x_k, equal to*

$$Q_x(x_k, x_k^i) = \int \Big[\log V_k(y_k - g(x_k)) + \log W_{k-1}(x_k - f(x_{k-1}))\Big] \times$$
$$W_{k-1}(x_k^i - f(x_{k-1}))p(x_{k-1}|\mathcal{Y}_{0:k-1})\, dx_{k-1}, \tag{2.3}$$

and its gradient is given by

$$\nabla_{x_k} Q_x(x_k, x_k^i) = \int \Big[\nabla_{x_k} \log V_k(y_k - g(x_k)) + \nabla_{x_k} \log W_{k-1}(x_k - f(x_{k-1}))\Big] \times$$
$$W_{k-1}(x_k^i - f(x_{k-1}))p(x_{k-1}|\mathcal{Y}_{0:k-1})dx_{k-1}. \tag{2.4}$$

Proof. We use Bayes' rule and Lemma 1, to yield

$$\begin{aligned} p(x_k|x_{k-1}, \mathcal{Y}_{0:k}) &= \frac{p(y_k|x_k, x_{k-1}, \mathcal{Y}_{0:k-1})p(x_k|x_{k-1}, \mathcal{Y}_{0:k-1})}{p(y_k|x_{k-1}, \mathcal{Y}_{0:k-1})}, \\ &= \frac{p(y_k|x_k)p(x_k|x_{k-1})}{p(y_k|x_{k-1})}. \end{aligned}$$

Thus,

$$\begin{aligned} p(x_k|x_{k-1}, \mathcal{Y}_{0:k}) &= \log p(y_k|x_k) + \log p(x_k|x_{k-1}) - \log p(y_k|x_{k-1}), \\ &= \log V_k(y_k - g(x_k)) + \log W_{k-1}(x_k - f(x_{k-1})) - \tilde{C}, \end{aligned} \tag{2.5}$$

where $\tilde{C} = \log p(y_k|x_{k-1})$ is independent from x_k and so integrates to an additive constant in the expectation inside (2.2).

Similarly, by Bayes' rule and Lemma 1,

$$\begin{aligned} p(x_{k-1}|x_k^i, \mathcal{Y}_{0:k}) &= p(x_{k-1}|x_k^i, \mathcal{Y}_{0:k-1}, y_k), \\ &= \frac{p(y_k|x_k^i, x_{k-1}, \mathcal{Y}_{0:k-1})p(x_{k-1}|x_k^i, \mathcal{Y}_{0:k-1})}{p(y_k|x_k^i, \mathcal{Y}_{0:k-1})}, \\ &= \frac{p(y_k|x_k^i)p(x_{k-1}|x_k^i, \mathcal{Y}_{0:k-1})}{p(y_k|x_k^i)}, \\ &= p(x_{k-1}|x_k^i, \mathcal{Y}_{0:k-1}). \end{aligned}$$

Applying Bayes' rule again,

$$p(x_{k-1}|x_k^i, \mathcal{Y}_{0:k}) = \frac{p(x_k^i|x_{k-1})p(x_{k-1}|\mathcal{Y}_{0:k-1})}{p(x_k^i|\mathcal{Y}_{0:k-1})}.$$

Hence,

$$\begin{aligned} p(x_{k-1}|x_k^i, \mathcal{Y}_{0:k}) &= p(x_{k-1}|x_k^i, \mathcal{Y}_{0:k-1}), \\ &= \frac{W_{k-1}(x_k^i - f(x_{k-1}))p(x_{k-1}|\mathcal{Y}_{0:k-1})}{\hat{C}}, \end{aligned} \tag{2.6}$$

where $\hat{C} = p(x_k^i|\mathcal{Y}_{0:k-1})$ is independent from x_{k-1} and leads to a positive multiplicative constant in the expectation inside (2.2).

Substituting (2.5) and (2.6) into (2.2) and neglecting the additive and positive multiplicative constants above yields

$$\begin{aligned} Q_x(x_k, x_k^i) = \int \Big\{ &[\log p(y_k|x_k) + \log p(x_k|x_{k-1})] \times \\ &p(x_k^i|x_{k-1})p(x_{k-1}|\mathcal{Y}_{0:k-1}) \Big\} dx_{k-1} \\ = \int \Big\{ &[\log V_k(y_k - g(x_k)) + \log W_{k-1}(x_k - f(x_{k-1}))] \times \\ &W_{k-1}(x_k^i - f(x_{k-1}))p(x_{k-1}|\mathcal{Y}_{0:k-1}) \Big\} dx_{k-1}. \end{aligned}$$

The smoothness assumptions in Section 2.2 permit exchanging differentiation and integration to arrive at the gradient formula.

$$\begin{aligned} \nabla_{x_k} Q_x(x_k, x_k^i) = \nabla_{x_k} \int \Big\{ &[\log V_k(y_k - g(x_k)) + \log W_{k-1}(x_k - f(x_{k-1}))] \times \\ &W_{k-1}(x_k^i - f(x_{k-1}))p(x_{k-1}|\mathcal{Y}_{0:k-1}) \Big\} dx_{k-1} \\ = \int \Big\{ &[\nabla_{x_k} \log V_k(y_k - g(x_k)) + \nabla_{x_k} \log W_{k-1}(x_k - f(x_{k-1}))] \times \\ &W_{k-1}(x_k^i - f(x_{k-1}))p(x_{k-1}|\mathcal{Y}_{0:k-1}) \Big\} dx_{k-1}. \end{aligned}$$

□

The gradient $\nabla_{x_k} Q_x(x_k, x_k^i)$ can be used within a gradient search algorithm to evaluate the M-step, which follows immediately.

2.3.2 M-step

The M-step maximizes the expectation in the E-step with respect to x_k, choosing x_k^{i+1} to be the maximizer,

$$x_k^{i+1} = \arg \max_{x_k \in \mathbb{R}^{r_x}} Q(x_k, x_k^i). \tag{2.7}$$

This choice of x_k^{i+1} will be shown in Theorem 2 to be non-decreasing in its likelihood compared to x_k^i. A proof of the general statement of EM algorithm can be found in [57]. We sketch the proof in the context of state filtering as opposed to the estimation of a constant parameter.

Theorem 2. *The sequence $\{x_k^i\}_{i \geq 0}$, commencing at $x_k^0 \in \mathbb{R}^{r_x}$ and satisfying the recursion (2.7), has the property that $\{\log p(x_k^i | \mathcal{Y}_{0:k})\}_{i \geq 0}$ is monotonically non-decreasing.*

Proof. The generic EM algorithm is posed with three variables: the unknown latent variable X, the measured data Y, and the unknown fixed parameter θ. In the context here, (X, Y, θ) are identified with $(x_{k-1}, \mathcal{Y}_{0:k}, x_k)$ and we can appeal to the proof of [57] for convergence. □

Corollary 1. *If the log-likelihood function $\log p(x_k | \mathcal{Y}_{0:k})$ is unimodal with the MLE being the only stationary point, then $\{x_k^i\}_{i \geq 0}$ converges to this MLE.*

This parallels Corollary 1 in [57].

The EM algorithm for ML state filtering at time-step k follows.

Algorithm 1

1: Given the current measurement y_k, the prior filtered density $p(x_{k-1}|\mathcal{Y}_{0:k-1})$ and initial time-k state estimate x_k^0, set $i \leftarrow 0$.

2: Evaluate the integral:

$$Q_x(x_k, x_k^i) = \int \Big\{ \Big[\log V_k(y_k - g(x_k)) + \log W_{k-1}(x_k - f(x_{k-1})) \Big] \times W_{k-1}(x_k^i - f(x_{k-1})) p(x_{k-1}|\mathcal{Y}_{0:k-1}) \Big\} dx_{k-1}. \tag{2.8}$$

3: Evaluate x_k^{i+1}:

$$x_k^{i+1} = \arg \max_{x_k \in \mathbb{R}^{r_x}} Q_x(x_k, x_k^i). \tag{2.9}$$

4: If $\{x_k^i\}_{i \geq 0}$ satisfies a convergence criterion: set $x_k^{\text{MLE}} = x_k^i$ and terminate. Else: $i \leftarrow i + 1$ and go to Step 2.

As guaranteed by Theorem 2, Algorithm 1 generates a sequence of estimates $\{x_k^i\}_{i \geq 0}$ which is non-decreasing in its likelihoods with respect to the filtered density.

With the generality of classes of functions f, g, W_k, and V_k implied by the problem formulation, the existence of an explicit solution to the M-step (2.7) cannot be concluded. And the user is left to pick a convenient maximization algorithm. However, gradient search algorithms, such as steepest ascent, can be a reasonable choice when the gradient exists. Its formula is given in Theorem 1, and its particle approximation in (2.11) below.

Although the densities V_k and W_k are assumed to be known, the integral (2.12) in Algorithm 1 requires the knowledge of $p(x_{k-1}|\mathcal{Y}_{0:k-1})$ as well. In general, outside the linear Gaussian case, such an integral cannot be analytically evaluated. Hence, a numerical approximation has to be considered. In this chapter, we turn to using a particle filter.

2.3.3 EM with a particle filter

The EM algorithm, so far, relies on the provision of the previous times' filtered density $p(x_{k-1}|\mathcal{Y}_{0:k-1})$ for the calculation in the E-step. A particle filter [13] provides a version of this information through a set of particles, i.e. a collection of values of $\{\tilde{x}^j_{k-1}\}_{j=1}^N$, and possibly their corresponding importance weights $\{\omega^j_{k-1|k-1}\}_{j=1}^N$, as illustrated in Figures 1 and 2. The density is approximated by

$$p(x_{k-1}|\mathcal{Y}_{0:k-1}) = \sum_{j=1}^{N} \omega^j_{k-1|k-1}\delta(x_{k-1} - \tilde{x}^j_{k-1}), \tag{2.10}$$

where $\delta(\cdot)$ is the Dirac delta function. If the particles are produced by resampling then each of the importance weights $\omega^j_{k-1|k-1} = 1/N$, even though the samples themselves can be repeated. Using these particles in (2.10), the expectation inside $Q_x(x_k, x^i_k)$ in (2.8) can be replaced by its particle approximation $\hat{Q}^N_x(x_k, x^i_k)$ in (2.12). And if a gradient search algorithm is to be used, the particle approximation of the gradient in (2.4) is

$$\nabla_{x_k}\hat{Q}^N_x(x_k, x^i_k) = \sum_{j=1}^{N} \Big[\nabla_{x_k} \log V_k(y_k - g(x_k)) + \nabla_{x_k} \log W_{k-1}(x_k - f(\tilde{x}^j_{k-1}))\Big] \times$$
$$W_{k-1}(x^i_k - f(\tilde{x}^j_{k-1}))\omega^j_{k-1|k-1}. \tag{2.11}$$

Algorithm 1 can then be recast using the particles for the E-step as follows.

Algorithm 2. EM State Filter (EMSF) Algorithm

1: Given y_k, the particles $\{\tilde{x}_{k-1}^j\}_{j=1}^N$, their normalized importance weights $\{\omega_{k-1|k-1}^j\}_{j=1}^N$ and initial time-k state estimate x_k^0, set $i \leftarrow 0$.

2: Evaluate the summation:

$$\hat{Q}_x^N(x_k, x_k^i) = \sum_{j=1}^{N} \Big[\log V_k(y_k - g(x_k)) + \log W_{k-1}(x_k - f(\tilde{x}_{k-1}^j)) \Big] \times$$
$$W_{k-1}(x_k^i - f(\tilde{x}_{k-1}^j))\omega_{k-1|k-1}^j. \tag{2.12}$$

3: Evaluate x_k^{i+1}:

$$x_k^{i+1} = \arg \max_{x_k \in \mathbb{R}^{r_x}} \hat{Q}_x^N(x_k, x_k^i). \tag{2.13}$$

4: If $\{x_k^i\}_{i \geq 0}$ satisfies a convergence criterion: set $x_k^{\text{MLE}} = x_k^i$ and terminate. Else: $i \leftarrow i + 1$ and go to Step 2.

Notice that

$$\begin{aligned} p(x_k|\mathcal{Y}_{0:k}) &= p(x_k|y_k, \mathcal{Y}_{0:k-1}), \\ &\propto p(y_k|x_k).p(x_k|\mathcal{Y}_{0:k-1}), \\ &\propto p(y_k|x_k) \int p(x_k|x_{k-1}) p(x_{k-1}|\mathcal{Y}_{0:k-1})\, dx_{k-1}. \end{aligned}$$

By using the particle approximation of the filtered density $p(x_{k-1}|\mathcal{Y}_{0:k-1})$, the following approximate density

$$p(x_k|\mathcal{Y}_{0:k}) = p(y_k|x_k) \sum_{j=1}^{N} p(x_k|\tilde{x}_{k-1}^j)\omega_{k-1|k-1}^j,$$

is the empirical filtered density [39], which Algorithm 2 seeks to maximize using EM. This density, which can be evaluated pointwise, can be used to compare the likelihoods of the convergence points, of multiple initializations, x_k^0s, of Algorithm 2. The multiple initializations

become necessary when the unimodality hypothesis of Corollary 1 fails to hold, as will be shown in the example in Subsection 2.5.2.

2.4 Gaussian state-space models with linear measurement equations

In the case of a nonlinear state-space model with linear measurement and additive Gaussian noises, Algorithm 2 reduces to a fixed-point iteration method which is convergent to a stationary point of the empirical filtered density. We use the term $\mathcal{N}(x|\mu, \Sigma)$ to denote the multivariate normal density of mean μ and covariance Σ evaluated at x.

Theorem 3. *For the nonlinear state-space system (3.1) with $g(x_k) = Hx_k$ for $H \in \mathbb{R}^{r_y \times r_x}$ and $w_k \sim \mathcal{N}(0, S_k)$, $v_k \sim \mathcal{N}(0, R_k)$, Steps 2 and 3 of Algorithm 2 reduce to the following fixed-point iteration.*

$$x_k^{i+1} = \frac{1}{\sum_{l=1}^{N} \lambda^l \omega_{k-1|k-1}^l} \times \Big[\sum_{j=1}^{N} f(\tilde{x}_{k-1}^j) \lambda^j \omega_{k-1|k-1}^j + \mathcal{B}_k (y_k - H \sum_{j=1}^{N} f(\tilde{x}_{k-1}^j) \lambda^j \omega_{k-1|k-1}^j) \Big], \tag{2.14}$$

where

$$\mathcal{B}_k = S_{k-1} H^T (H S_{k-1} H^T + R_k)^{-1},$$

and

$$\lambda^j = \mathcal{N}(x_k^i - f(\tilde{x}_{k-1}^j)|0, S_{k-1}), \quad j = 1, 2, \ldots N. \tag{2.15}$$

Proof. The summation (2.12) in Algorithm 2 reduces to

$$\hat{Q}_x^N(x_k, x_k^i) = \sum_{j=1}^{N} \Big[-(y_k - g(x_k))^T R_k^{-1}(y_k - g(x_k)) \\ -(x_k - f(\tilde{x}_{k-1}^j))^T S_{k-1}^{-1}(x_k - f(\tilde{x}_{k-1}^j)) \Big] \times \lambda^j \omega_{k-1|k-1}^j$$

which is quadratic in x_k. Hence, the unique maximizer of $Q_x^N(x_k, x_k^i)$ can be found by setting $\nabla_{x_k} Q_x^N(x_k, x_k^i)$ to zero. Thus,

$$x_k^{i+1} = \Big[H^T R_k^{-1} H + S_{k-1}^{-1} \Big]^{-1} \frac{1}{\sum_{l=1}^N \lambda^l \omega_{k-1|k-1}^l} \times \\ \Big\{ H^T R_k^{-1} y_k + S_{k-1}^{-1} \sum_{j=1}^{N} f(\tilde{x}_{k-1}^j) \lambda^j \omega_{k-1|k-1}^j \Big\}.$$

Applying the matrix inversion lemma completes the proof. □

We have the following ML state filtering algorithm.

Algorithm 3. EMSF Algorithm for systems with linear measurements and additive Gaussian noises

1: Subject to the assumptions of Theorem 3 and given: y_k, the particles $\{\tilde{x}_{k-1}^j\}_{j=1}^N$, their normalized importance weights $\{\omega_{k-1|k-1}^j\}_{j=1}^N$ and initial time-k state estimate x_k^0, set $i \leftarrow 0$.

2: Update x_k^i to x_k^{i+1} per (2.14) of Theorem 3.

3: If $\{x_k^i\}_{i \geq 0}$ satisfies a convergence criterion: set $x_k^{\text{MLE}} = x_k^i$ and terminate. Else: $i \leftarrow i + 1$ and go to step 2.

2.5 ML state prediction and smoothing by the expectation maximization algorithm

The EMSF algorithm uses the previous time's particle filtered density and the current measurement to produce the estimate. ML state prediction and smoothing rely on different

particle densities, and for smoothing several such densities plus the current measurement. These data are used in fashions distinct from EMSF, as we now explain.

2.5.1 Prediction

Given,

- the particle approximation of the time-m filtered density $p(x_m|\mathcal{Y}_{0:m})$, for $m \in \{1, \ldots, k-1\}$, i.e.
$$p(x_m|\mathcal{Y}_{0:m}) = \sum_{j=1}^{N} \omega_{m|m}^{j} \delta(x_m - \tilde{x}_{m|m}^{j}).$$
- initial state estimate x_k^0,

the prediction problem is to find the MLE of the state-vector x_k using the log-likelihood function $\log p(x_k|\mathcal{Y}_{0:m})$ for $m < k$.

The time-update of the bootstrap particle filter can be used to propagate the particles and achieve the particle approximation

$$p(x_{k-1}|\mathcal{Y}_{0:m}) = \sum_{j=1}^{N} \omega_{m|m}^{j} \delta(x_{k-1} - \tilde{x}_{k-1|m}^{j}). \tag{2.16}$$

The importance weights are unchanged because no extra measurements are available after time-m. The EM algorithm can then be used to evaluate the MLE of x_k of the predicted density $p(x_k|\mathcal{Y}_{0:m})$, using the particle approximation (2.16). The E-step is performed by evaluating the following expectation

$$\begin{aligned} Q_x^p(x_k, x_k^i) &= \mathbb{E}\left\{\log p(x_k|x_{k-1}, \mathcal{Y}_{0:m})|x_k^i, \mathcal{Y}_{0:m}\right\}, \\ &= \int \left\{ \log p(x_k|x_{k-1}, \mathcal{Y}_{0:m}) p(x_{k-1}|x_k^i, \mathcal{Y}_{0:m}) \right\} dx_{k-1}, \end{aligned}$$

which, analogous to the proof of Theorem 1, is equivalent to

$$Q_x^p(x_k, x_k^i) = \int \left\{ [\log W_{k-1}(x_k - f(x_{k-1}))] \times W_{k-1}(x_k^i - f(x_{k-1})) p(x_{k-1}|\mathcal{Y}_{0:m}) \right\} dx_{k-1}. \tag{2.17}$$

The particle approximation (2.16) can be used to construct $Q_x^{p,N}$, an approximant of (2.17) up to additive and multiplicative constants, which are independent from x_k,

$$\hat{Q}_x^{p.N}(x_k, x_k^i) = \sum_{j=1}^{N} \left[\log W_{k-1}(x_k - f(\tilde{x}_{k-1|m}^j)) \right] \times W_{k-1|m}(x_k^i - f(\tilde{x}_{k-1|m}^j)) \omega_{m|m}^j.$$

The M-step is performed by maximizing this approximant over x_k. Hence, the EM algorithm for ML state prediction,

Algorithm 4. EM State Predictor (EMSP) Algorithm

1: Given the particles $\{\tilde{x}_{k-1|m}^j\}_{j=1}^N$, their normalized importance weights $\{\omega_{m|m}^j\}_{j=1}^N$ and initial time-k state estimate x_k^0, set $i \leftarrow 0$.

2: Evaluate the summation:

$$\hat{Q}_x^{p.N}(x_k, x_k^i) = \sum_{j=1}^{N} \left[\log W_{k-1}(x_k - f(\tilde{x}_{k-1|m}^j)) \right] \times W_{k-1}(x_k^i - f(\tilde{x}_{k-1|m}^j)) \omega_{m|m}^j. \tag{2.18}$$

3: Evaluate x_k^{i+1}:

$$x_k^{i+1} = \arg \max_{x_k \in \mathbb{R}^{r_x}} \hat{Q}_x^{p,N}(x_k, x_k^i). \tag{2.19}$$

4: If $\{x_k^i\}_{i \geq 0}$ satisfies a convergence criterion: set $x_k^{\text{MLE}} = x_k^i$ and terminate. Else: $i \leftarrow i + 1$ and go to step 2.

Similar versions of Theorem 2 and Corollary 1 hold for the sequence $\{x_k^i\}_{i \geq 0}$ generated by Algorithm 4, i.e. the sequence $\{\log p(x_k^i|\mathcal{Y}_m)\}_{i \geq 0}$ is non-decreasing, and if $\log p(x_k|\mathcal{Y}_m)$ is unimodal and its MLE is the only stationary point, then $\{x_k^i\}_{i \geq 0}$ of Algorithm 4 converges to the

MLE.

2.5.2 Smoothing

In contrast to prediction and filtering, smoothed state estimation involves considerably greater complexity and, in the MLE environment, greater demands of the particle filter. This leads to an EM algorithm which has complexity $\mathcal{O}(N^2)$. The algorithm EMSS, however, follows directly, as we now show.

Given,

- the measurements sequence $\mathcal{Y}_{0:n}$, for $n > k$,
- The particle approximations of the following densities:

$$p(x_{k-1}|\mathcal{Y}_{0:k-1}) = \sum_{j=1}^{N} \omega_{k-1|k-1}^{j} \delta(x_{k-1} - \tilde{x}_{k-1|k-1}^{j}), \tag{2.20}$$

$$p(x_k|\mathcal{Y}_{0:k}) = \sum_{j=1}^{N} \omega_{k|k}^{j} \delta(x_k - \tilde{x}_{k|k}^{j}), \tag{2.21}$$

$$p(x_{k+1}|\mathcal{Y}_{0:n}) = \sum_{j=1}^{N} \omega_{k+1|n}^{j} \delta(x_{k+1} - \tilde{x}_{k+1|n}^{j}), \tag{2.22}$$

which can be supplied by a forward–backward particle smoother [13],

- initial state estimate x_k^0,

the smoothing problem is to find the MLE of the state-vector x_k using the log-likelihood function $\log p(x_k|\mathcal{Y}_{0:n})$ for $n > k + 1$.

The EM algorithm can be used to evaluate the MLE of x_k of the smoothed density

$p(x_k|\mathcal{Y}_{0:n})$. The E-step is performed by evaluating

$$\begin{aligned} & Q_x^S(x_k, x_k^i) \\ & = \mathbb{E}\left\{\log p(x_k|x_{k+1}, \mathcal{Y}_{0:n})|x_k^i, \mathcal{Y}_{0:n}\right\}, \\ & = \int \log p(x_k|x_{k+1}, \mathcal{Y}_{0:n}) p(x_{k+1}|x_k^i, \mathcal{Y}_{0:n})\, dx_{k+1}. \end{aligned} \tag{2.23}$$

Where in (2.23), using Bayes' rule 3 times and Lemma 1, the density

$$\begin{aligned} p(x_k|x_{k+1}, \mathcal{Y}_{0:n}) &= \frac{p(\mathcal{Y}_{k+1:n}|x_{k+1}, x_k, \mathcal{Y}_{0:k}) p(x_k|x_{k+1}, \mathcal{Y}_{0:k})}{p(\mathcal{Y}_{k+1:n}|x_{k+1}, \mathcal{Y}_{0:k})}, \\ &= \frac{p(\mathcal{Y}_{k+1:n}|x_{k+1}) p(x_k|x_{k+1}, \mathcal{Y}_{0:k})}{p(\mathcal{Y}_{k+1:n}|x_{k+1})}, \\ &= p(x_k|x_{k+1}, \mathcal{Y}_{0:k}), \\ &= \frac{p(x_{k+1}|x_k) p(x_k|\mathcal{Y}_{0:k})}{p(x_{k+1}|\mathcal{Y}_{0:k})}, \\ &= \frac{p(x_{k+1}|x_k) p(y_k|x_k) p(x_k|\mathcal{Y}_{0:k-1})}{p(x_{k+1}|\mathcal{Y}_{0:k}) p(y_k|\mathcal{Y}_{0:k-1})}. \end{aligned}$$

The term $p(y_k|\mathcal{Y}_{0:k-1})$ is independent from x_k and leads to an additive constant in the expectation inside (2.23). Thus,

$$\begin{aligned} & \log p(x_k|x_{k+1}, \mathcal{Y}_{0:n}) \\ & \quad \propto \log p(x_{k+1}|x_k) + \log p(y_k|x_k) + \log p(x_k|\mathcal{Y}_{0:k-1}), \\ & \quad \propto \log W_k(x_{k+1} - f(x_k)) \\ & \quad\quad + \log V_k(y_k - g(x_k)) + \log p(x_k|\mathcal{Y}_{0:k-1}). \end{aligned} \tag{2.24}$$

The other density in (2.23), using the same conditional independence result and Bayes'

rule,

$$
\begin{aligned}
p(x_{k+1}|x_k^i, \mathcal{Y}_{0:n}) &= \frac{p(x_k^i|x_{k+1}, \mathcal{Y}_{0:n})p(x_{k+1}|\mathcal{Y}_{0:n})}{p(x_k^i|\mathcal{Y}_{0:n})}, \\
&= \frac{p(x_k^i|x_{k+1}, \mathcal{Y}_{0:k})p(x_{k+1}|\mathcal{Y}_{0:n})}{p(x_k^i|\mathcal{Y}_{0:n})}, \\
&= \frac{p(x_{k+1}|x_k^i)p(x_k^i|\mathcal{Y}_{0:k})p(x_{k+1}|\mathcal{Y}_{0:n})}{p(x_{k+1}|\mathcal{Y}_{0:k})p(x_k^i|\mathcal{Y}_{0:n})},
\end{aligned} \tag{2.25}
$$

where the terms $p(x_k^i|\mathcal{Y}_{0:k})$ and $p(x_k^i|\mathcal{Y}_{0:n})$ are independent from x_{k+1} and yield a positive multiplicative constant in the expectation inside (2.23). Hence,

$$
\begin{aligned}
p(x_{k+1}|x_k^i, \mathcal{Y}_{0:n}) &\propto \frac{p(x_{k+1}|x_k^i)p(x_{k+1}|\mathcal{Y}_{0:n})}{p(x_{k+1}|\mathcal{Y}_{0:k})}, \\
&\propto \frac{W_k(x_{k+1} - f(x_k^i))p(x_{k+1}|\mathcal{Y}_{0:n})}{p(x_{k+1}|\mathcal{Y}_{0:k})}.
\end{aligned} \tag{2.26}
$$

Substituting (2.24) and (2.26) in (2.23), after dropping the additive and the multiplicative constants, results in

$$
\begin{aligned}
Q_x^S(x_k, x_k^i) = \log V_k(y_k - g(x_k)) + \log p(x_k|\mathcal{Y}_{0:k-1}) + \\
\int \left\{ [\log W_k(x_{k+1} - f(x_k))] \times \frac{W_k(x_{k+1} - f(x_k^i))p(x_{k+1}|\mathcal{Y}_{0:n})}{p(x_{k+1}|\mathcal{Y}_{0:k})} \right\} dx_{k+1}.
\end{aligned} \tag{2.27}
$$

Notice that, for the construction of (2.27), the densities $p(x_k|\mathcal{Y}_{0:k-1})$, $p(x_{k+1}|\mathcal{Y}_{0:n})$ and $p(x_{k+1}|\mathcal{Y}_{0:k})$ are required. The particle calculation (2.21) can be used to approximate $p(x_{k+1}|\mathcal{Y}_{0:k})$:

$$
p(x_{k+1}|\mathcal{Y}_{0:k}) = \int p(x_{k+1}|x_k)p(x_k|\mathcal{Y}_{0:k})\, dx_k = \sum_{j=1}^{N} \omega_{k|k}^j W_k(x_{k+1} - f(\tilde{x}_{k|k}^j)). \tag{2.28}
$$

Similarly, the density $p(x_k|\mathcal{Y}_{0:k-1})$ can be approximated using (2.20),

$$\log p(x_k|\mathcal{Y}_{0:k-1}) = \log \sum_{j=1}^{N} \omega_{k-1|k-1}^{j} W_{k-1}(x_k - f(\tilde{x}_{k-1|k-1}^{j})). \tag{2.29}$$

Finally, by using (2.29) and (2.28) in (2.27), and (2.22) to approximate the integral term in (2.27), after neglecting additive and multiplicative constants, we obtain $\hat{Q}_x^{S.N}$, the particle approximation of Q_x^S,

$$\hat{Q}_x^{S.N}(x_k, x_k^i) = \log V_k(y_k - g(x_k)) + \log \sum_{j=1}^{N} \omega_{k-1|k-1}^{j} W_{k-1}(x_k - f(\tilde{x}_{k-1|k-1}^{j})) + \sum_{t=1}^{N} \Big[\log W_k(\tilde{x}_{k+1|n}^{t} - f(x_k)) \Big] \times \frac{W_k(\tilde{x}_{k+1|n}^{t} - f(x_k^i))}{\sum_{d=1}^{N} \omega_{k|k}^{d} W_k(\tilde{x}_{k+1|n}^{t} - f(\tilde{x}_{k|k}^{d}))} \omega_{k+1|n}^{t}. \tag{2.30}$$

The EM algorithm for ML state smoothing,

Algorithm 5. EM State Smoother (EMSS) Algorithm

1: Given the sets of particles $\{\tilde{x}_{k-1|k-1}^{j}\}_{j=1}^{N}$, $\{\tilde{x}_{k|k}^{j}\}_{j=1}^{N}$, $\{\tilde{x}_{k+1|n}^{j}\}_{j=1}^{N}$, their corresponding normalized importance weights, and initial time-k state estimate x_k^0, set $i \leftarrow 0$.

2: Evaluate:

$$\hat{Q}_x^{S.N}(x_k, x_k^i) = \log V_k(y_k - g(x_k)) + \log \sum_{j=1}^{N} \omega_{k-1|k-1}^{j} W_{k-1}(x_k - f(\tilde{x}_{k-1|k-1}^{j})) + \sum_{t=1}^{N} \Big[\log W_k(\tilde{x}_{k+1|n}^{t} - f(x_k)) \Big] \times \frac{W_k(\tilde{x}_{k+1|n}^{t} - f(x_k^i))}{\sum_{d=1}^{N} \omega_{k|k}^{d} W_k(\tilde{x}_{k+1|n}^{t} - f(\tilde{x}_{k|k}^{d}))} \omega_{k+1|n}^{t}. \tag{2.31}$$

3: Evaluate x_k^{i+1}:

$$x_k^{i+1} = \arg \max_{x_k \in \mathbb{R}^{r_x}} \hat{Q}_x^{S,N}(x_k, x_k^i). \tag{2.32}$$

4: If $\{x_k^i\}_{i \geq 0}$ satisfies a convergence criterion: set $x_k^{\text{MLE}} = x_k^i$ and terminate. Else: $i \leftarrow i + 1$ and go to step 2.

2.6 Examples

We apply EMSF to the following computed examples, both of which satisfy the structure of Theorem 3.

2.6.1 Example 1: Linear Gaussian state-space model

In the linear Gaussian state-space model case, all random variables under consideration are Gaussian. Hence, the conditional mean is equivalent to the conditional mode [41]. In the following example, the Kalman filter, which is designed to track the propagation of the conditional mean, is compared the EMSF. It will be shown that the EMSF, starting from a random initialization x_k^0, converges to the Kalman Filter solution, which, for known densities, is guaranteed by Corollary 1.

Consider the following model,

$$\begin{aligned} x_{k+1} &= Fx_k + w_k, \\ y_k &= Hx_k + v_k, \\ v_k &\sim \mathcal{N}(0, R),\ w_k \sim \mathcal{N}(0, S), \\ x_0 &\sim \mathcal{N}(x_{0|-1}, \Sigma_{0|-1}), \end{aligned} \tag{2.33}$$

where: $x_{0|-1}$ and $\Sigma_{0|-1}$ are respectively the mean and the covariance matrix of $x_{0|-1}$ prior to receiving measurements, the sequences $\{w_k\}$ and $\{v_k\}$ and the random variable x_0 follow the independence and whiteness assumptions as in Section 2.2.

We apply Algorithm 3, yielding

$$\lambda^j = W_{k-1}(x_k^i - F\tilde{x}_{k-1}^j) = \mathcal{N}(x_k^i - F\tilde{x}_{k-1}^j|0, S),$$
$$x_k^{i+1} = \left[H^T R^{-1} H + S^{-1}\right]^{-1} \frac{1}{\sum_{l=1}^N \lambda^l \omega_{k-1|k-1}^l} \times \left\{H^T R^{-1} y_k + S^{-1} F \sum_{j=1}^N \tilde{x}_{k-1}^j \lambda^j \omega_{k-1|k-1}^j\right\}. \tag{2.34}$$

Or, using the matrix inversion lemma,

$$x_k^{i+1} = \left[F \sum_{j=1}^N \tilde{x}_{k-1}^j \lambda^j \omega_{k-1|k-1}^j + \mathcal{B}(y_k - HF \sum_{j=1}^N \tilde{x}_{k-1}^j \lambda^j \omega_{k-1|k-1}^j)\right] \frac{1}{\sum_{l=1}^N \lambda^l \omega_{k-1|k-1}^l}, \tag{2.35}$$

with $\mathcal{B} = SH^T(HSH^T + R)^{-1}$. Note that (2.35) bears a passing resemblance to the Kalman filter recursion as does (2.34) to the information filter.

Now take

$$x_k = \begin{bmatrix} x_{k,1} \\ x_{k,2} \\ x_{k,3} \end{bmatrix}, \; F = \begin{bmatrix} 0.66 & -1.31 & -1.11 \\ 0.07 & 0.73 & -0.06 \\ 0.00 & 0.08 & 0.80 \end{bmatrix},$$
$$H = \begin{bmatrix} 0 & 1 & 1 \end{bmatrix}, S = \begin{bmatrix} 0.2 & 0 & 0 \\ 0 & 0.3 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}, R = 0.1,$$
$$x_{0|-1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \; \Sigma_{0|-1} = \begin{bmatrix} 0.3 & 0 & 0 \\ 0 & 0.3 & 0 \\ 0 & 0 & 0.3 \end{bmatrix}.$$

A simulation over 100 time-steps, with $N = 2000$ particles, is conducted with the results plotted in Figure 2.3. The recursion (2.35) at each time-step k is initialized by $x_k^0 \sim \mathcal{N}(0, I_{3\times 3})$. The results show that the EMSF converges to the Kalman Filter solution. The Kalman Filter

solution, i.e. the conditional mean, is also the conditional mode for linear Gaussian state-space models. Therefore, by Corollary 1, the EMSF solution and the Kalman Filter solution should match, subject to the adequacy of the particle filter, which is shown by this example.

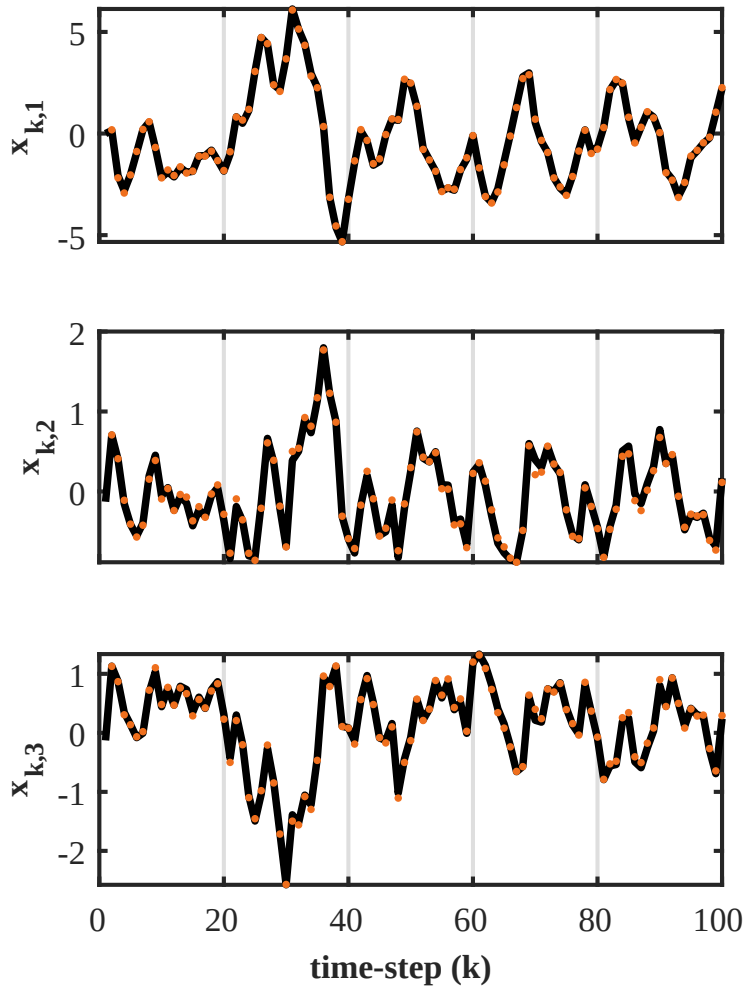

Figure 2.3. The Kalman Filter solution in black, and the EMSF solution in orange dots.

Figure 2.4 depicts the convergence of the iterates at three individual times of $x_{k,1}$. The algorithm stops whenever the maximum absolute relative error is below 0.005, i.e.

$$\max_{q\in\{1,2,3\}} |(x_{k,q}^{i+1} - x_{k,q}^{i})/x_{k,q}^{i+1}| \leq 0.5\%.$$

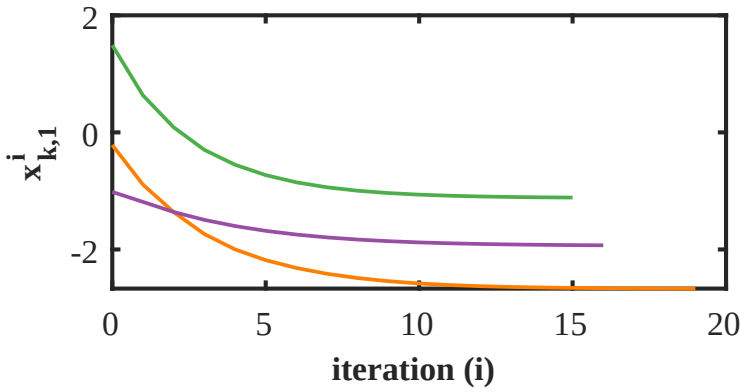

Figure 2.4. The convergence of the sequences $\{x^i_{k,1} : i \geq 0\}$ for $k = 15, 55, 90$ in green, orange and purple, respectively.

The EMSF algorithm is of $\mathcal{O}(N)$ computational complexity, per iteration, with N the number of particles. Ten simulations were conducted using MATLAB, an uncompiled interpretive program, on a computer with a 2.5 GHz Intel Core i5-7200U processor and 8.00 GB of RAM. The running times for the one hundred time points averaged over the ten runs were: 3 milliseconds for the Kalman filter, 3.4 seconds for the bootstrap particle filter, and 26.4 seconds for EMSF including the bootstrap particle filter.

2.6.2 Example 2: Nonlinear state-space model with multimodal/skewed state distribution

In this example, the EMSF is tested with a nonlinear state-space model with highly skewed and/or multimodal state densities. The EMSF solution is compared to the conditional mean of the bootstrap particle filter. This example also conforms to the conditions of Theorem 3.

Consider the following nonlinear state-space model,

$$\begin{aligned} x_{k+1} &= \alpha_k \tanh(\pi x_k) + w_k, \\ y_k &= \frac{1}{2} x_k + v_k, \\ w_k &\sim \mathcal{N}(0, \frac{1}{5}),\ v_k \sim \mathcal{N}(0,1), \\ x_0 &\sim \mathcal{N}(0,1), \end{aligned} \tag{2.36}$$

where $\alpha_k = (1 + 0.5\sin(2\pi k/20))$. Theorem 3 yields the following recursion

$$x_k^{i+1} = \frac{1}{10.5}\frac{1}{\sum_{l=1}^{N}\lambda^l\omega_{k-1|k-1}^l}\times \sum_{l=1}^{N}[y_k + 10\alpha_k\tanh(\pi\tilde{x}_{k-1}^j)\lambda^j\omega_{k-1|k-1}^j], \tag{2.37}$$

where $\lambda^j = \mathcal{N}(x_k^i - \alpha_{k-1}\tanh(\pi\tilde{x}_{k-1}^j)|0, 1/5)$.

The simulation is carried out over 40 time-steps, and $N = 500$ particles. The EMSF, at each time k, is defined by the recursion (2.37) and is initialized by five independent initial guesses $x_k^0 \sim \mathcal{U}[-2, 2]$. The number of iterations is set to be ≤ 10, and the point of highest final likelihood is taken over the five initial conditions. For this example, the empirical filtered density at each time step k is

$$p(x_k|\mathcal{Y}_{0:k}) = \mathcal{N}(y_k - \frac{1}{2}x_k|0, 1)\times \sum_{j=1}^{N}\mathcal{N}(x_k - \alpha_{k-1}\tanh(\pi\tilde{x}_{k-1}^j)|0, \frac{1}{5})\omega_{k-1|k-1}^j,$$

where the particles are $\{\tilde{x}_{k-1}^j\}$ and their weights are $\{\omega_{k-1|k-1}^j\}$. This density can be used to compare the likelihoods of the convergence points resulting from multiple initial guesses. The results are shown in Figures 2.5 to 2.7.

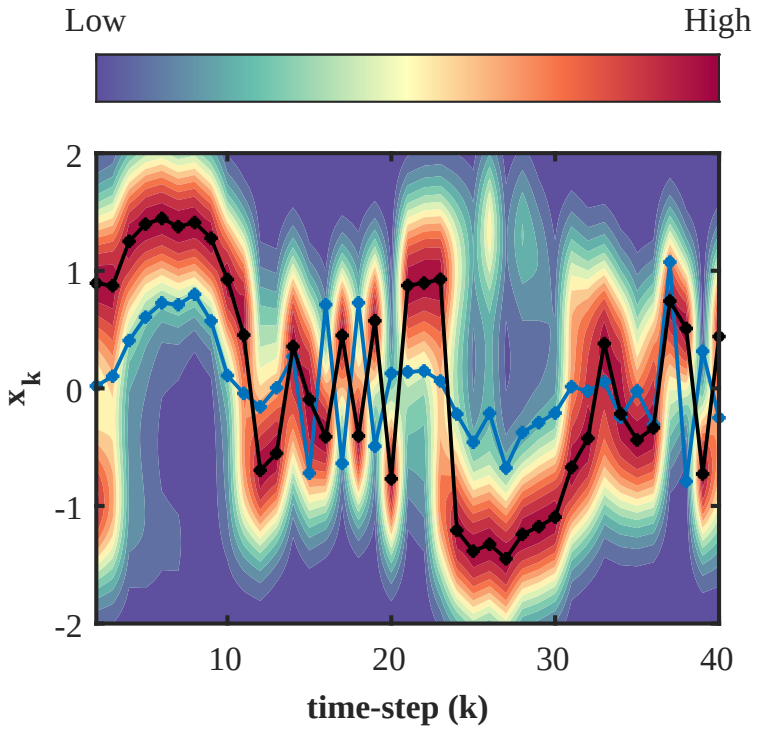

Figure 2.5. The coloring represents the empirical density function (unnormalized). In black is the output of the EMSF algorithm, and in blue is the conditional mean of the bootstrap particle filter.

Each vertical time-section of the coloring in Figure 2.5 is the unnormalized empirical filtered density. The EMSF ML estimate is shown in black. The conditional mean from the particle filter, computed by averaging, is shown in blue. The EMSF is able to track the ML estimate in spite of multimodality and skewness of the underlying density. The simulation in this figure uses 500 particles. It takes 1.8 seconds on a computer with a 2.5 GHz Intel Core i5-7200U processor and 8.00 GB of RAM running MATLAB.

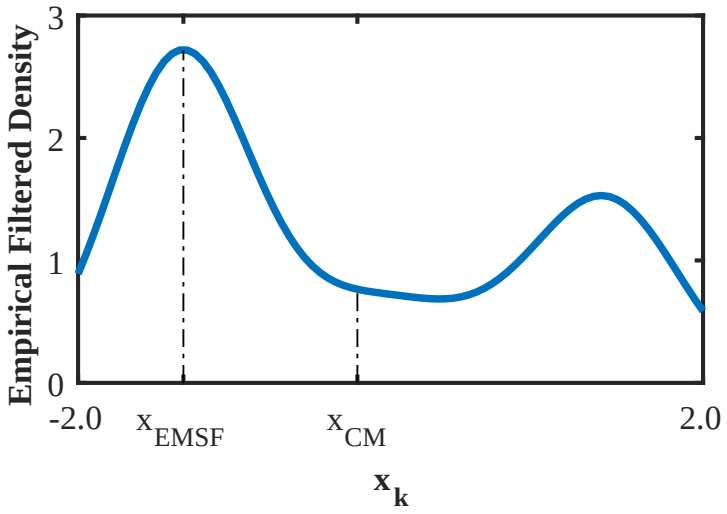

Figure 2.6. The bi-modal empirical filtered density function of x_k at $k = 26$ (apart from a multiplicative constant). x_{EMSF} is the solution of the EMSF algorithm and x_{CM} is the conditional mean of the bootstrap particle filter.

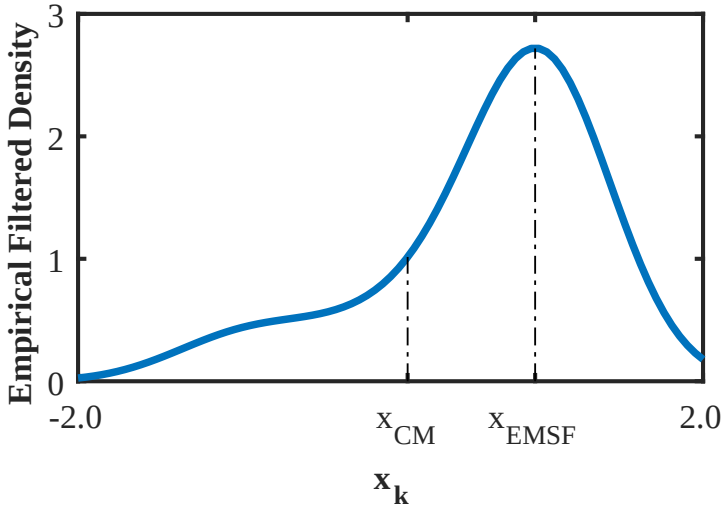

Figure 2.7. The highly skewed empirical filtered density function of x_k at $k = 10$. x_{EMSF} is the solution of the EMSF algorithm and x_{CM} is the conditional mean of the bootstrap particle filter.

Figures 2.6 and 2.7 show the unnormalized empirical filtered densities at different times from this simulation. The EMSF estimate, x_{EMSF}, in both cases, converges to the modes of these densities while the conditional mean is a less useful statistic from the particle filter for ML estimation because of multimodality and skewness.

2.7 Conclusion

This chapter contributes a novel approach to incorporate the EM algorithm, along with a particle filter, into the problem of maximum likelihood state estimation. Algorithms for maximum likelihood state filtering, prediction, and smoothing are presented, together with the desirable convergence properties they inherit from the EM algorithm and the bootstrap particle filter. Additionally, for the wide range of nonlinear state-space systems possessing linear measurements with additive Gaussian noises, the filtering and prediction algorithms reduce to derivative-free optimization through a fixed-point iteration. The EMSF algorithm is tested on two examples:

1. The system in the first example is linear third-order Gaussian. In comparison with the Kalman filter, this example demonstrates:
    - the concordance of the estimates;
    - the computational simplicity of the Kalman filter;
    - the computational feasibility of the EMSF combined with its attendant particle filter.
2. The second example demonstrates:
    - the ability of the EMSF to estimate the MLE of a state density which exhibits multimodality and strong skewness;
    - the simplicity of the EMSF algorithm after it is reduced to a fixed-point iteration.

Further work is being pursued to understand the role of MLE state estimates in constrained control, such as Model Predictive Control where linearity and Gaussian assumptions can contravene the problem formulation.

Acknowledgement

Chapter 2, in full, is a published material as it appears in: Mohammad S. Ramadan, Robert R. Bitmead “Maximum Likelihood recursive state estimation using the Expectation

Maximization algorithm." Automatica, October 2022. The dissertation author was the primary investigator and author of this paper.

Chapter 3

State estimation for control: an approach for output-feedback stochastic MPC

Abstract

This chapter provides a new approach to the determination of a single state value for stochastic output feedback problems using paradigms from Model Predictive Control, particularly the distinction between open-loop and closed-loop control and between deterministic optimal control and stochastic optimal control. The State Selection Algorithm is presented and relies on given dynamics and constraints, a nominal deterministic state-feedback controller, and a sampling based method to select the best state value, based on optimizing a prescribed finite-horizon performance function, over the available candidates provided by a particle filter. The cost function is minimized over the horizon with controls determined by the nominal controller and the selected states. So, the minimization is performed not over the selection of the control other than through the choice of state value to use. The algorithm applies generally to nonlinear stochastic systems and relies on Monte Carlo sampling and averaging. However, in linear quadratic polyhedrally constrained cases the technique reduces to a quadratic program for the state value. The algorithm is evaluated in a set of computational examples, which illustrate its efficacy and limitations. Numerical aspects and the opportunity for parallelization are discussed. The examples demonstrate the algorithm operating, in closed-loop with its attendant particle filter, over the long horizon.

3.1 Introduction

We consider state selection for output feedback control of nonlinear stochastic systems. Starting with a filtered particle density of the state and a nominal control law, an algorithm is presented to choose a single state value from the density, which is then used in the control. The algorithm proceeds from the repeated selection of a single particle followed by a closed-loop stage, which generates a set of candidate control sequences, which then are evaluated for open-loop performance averaged over all the particles. This appearance of open-loop and closed-loop calculations highlights a significant distinction between deterministic and stochastic optimal feedback control, as is the propagation of single estimates versus whole densities. Both are discussed shortly. This also brings to the fore a difficulty in the formulation of output feedback stochastic MPC, which provides a touchstone control problem from which to appreciate the methods and, in part, their genealogy.

Context and MPC

The surge of interest and research in Model Predictive Control (MPC) since the late 1980s is in large measure attributable to its capacity to handle constraints [32]. It does this within a receding-horizon optimal control context, which solves repeatedly a constrained open-loop optimal control problem from the current state, presumed to be available. That is, MPC is formulated as full-state feedback with the feedback being achieved by the receding horizon device of applying only the first element of the open-loop control sequence, each element of which is a function of the initial state, before measuring the next state and re-solving the open-loop problem. Being open-loop, each of these constrained optimization problems is manageable in its complexity.

MPC using partially observed states enters into the realms of stochastic optimal control, which is known to be computationally intractable in all but the simplest of cases [15, 5], with or without constraints. Further, the presence of an equivalent open-loop optimal solution sequence

for these problems evanesces; optimal controls are necessarily feedback only [21]. Our study here is to walk a middle path between the in computational intractability (and optimality) of stochastic optimal control and the convenience of open-loop methods. Given a specific state feedback control law, $u_k = \kappa(x_k)$, as might be provided, say, by explicit MPC, we develop an heuristic (and evidently suboptimal) approach to the selection of a single state value from the state particle filtered conditional density. The control law κ thus effected relies on the filtered density solely through the selection process. For unconstrained Linear Quadratic problems, the selection of the conditional mean of the particle density would be optimal [3]. However for constrained and/or nonlinear problems, state selection depends on the given control law and evaluation, in open-loop and on the particle ensemble, of both constraint satisfaction and a performance measure. The evaluation of candidate state vectors builds on the finite-horizon open-loop nature of the MPC iteration and combines two aspects along a prediction horizon: the probability of constraint violation and the calculation of a predicted performance function, both averaged over the particle density, as is explicated later.

Stochastic optimal control with partially observed state, which we abbreviate to *stochastic optimal control*, is solved using Stochastic Dynamic Programming [21, 8] which requires propagation of full conditional densities at each time. The optimal control law necessarily is feedback-only; it cannot be precomputed along a horizon because it depends on anticipated measurements and the associated conditional density propagation. This should be compared to deterministic optimal control (and implicitly MPC methods) where the open-loop optimal control can be computed, say via Pontryagin's Maximum Principle, and coincides as a function of time with the feedback optimal control of dynamic programming. It is this feature of deterministic optimal control, which receding-horizon MPC relies upon for the generation of a feedback control law from an open-loop constrained optimization. The optimization, since it is open-loop, is computationally achievable. These concepts of open- and closed-loop underpin the computational issues in stochastic optimal control.

Because of the proscriptive burden of stochastic optimal control, several less cumbersome

but approximate and suboptimal approaches to these problems have been developed.

- ✻ Wide-sense approaches, in which state uncertainty is presented by approximates of its first two central moments, and an approximate dynamics by linearization or Taylor expansion, of the propagation of these moments with the input and anticipated measurements, is used [5].

- ✻ Explicit probing, that is, including a state variance-based expression explicitly in the cost of a deterministic version of the stochastic problem, as in [22].

- ✻ Stochastic MPC approaches in which active learning, or the reliance on future measurements, is typically dropped. This permits achieving an open-loop solution. Example stochastic MPC approaches that are designed to deal with initial state uncertainty include:

  - ✯ a modified version of scenario-based approaches directed towards initial state uncertainty [47],

  - ✯ sequential Monte Carlo approaches for sampling the input space, for nonlinear but unconstrained problems [19],

  - ✯ stochastic tube approaches [58, 9, 16], which by their formulation are limited in applicability to linear systems.

A more complete review of stochastic MPC approaches can be found in [35]. Here, we present an algorithm akin to the stochastic MPC approaches (or equivalently to open-loop feedback control [8]). Differently from these approaches, we seek to select the initial state value rather than designing the controller. At variance from stochastic tube approaches, our formulation extends naturally to constrained nonlinear dynamics. This is due to the sampling approach we use, which, in contrast to [19], is parallelizable over the samples.

As discussed by Striebel in [50], purpose or function should guide data reduction efforts. Our approach is to focus on the selection of a single state vector value from the ensemble of

particles representing the density. This reduction is influenced by a stated control objective, with a given – as yet, neither necessarily stabilizing, nor optimal nor constraint-feasible – nominal control law. So the control value, guided by the control objective, is indirectly determined by the nominal law and the selection process. This differs from the other methods which directly select the control value. The method proceeds from the following information: the system state equation with white process noise, a nominal state-feedback control law, the constraints, the noise density, a filtered particle state density, a cost function along the prediction horizon, state and input constraints, and their violation rate tolerance. Running alongside this control computation is the propagation of the filtered state particle density, whose properties also affect the closed-loop behavior.

The incorporation of the initial state as a decision variable in MPC was used previously in the literature, for linear dynamics with full state-feedback. In [45], the initial state is augmented to the decision variable of a tube-based MPC problem, to increase the likelihood of establishing feasibility. Also, in [33], the previous concept of augmentation is used, but to achieve constant value function over an invariant set around the origin. our approach does not assume a known initial state and extends naturally to nonlinear dynamics.

In the linear problem with linear nominal feedback control, quadratic cost and polyhedral constraints, the solution developed by the state selection approach coincides with a quadratic program on the initial states.

Outline

The precise problem formulation and our proposed algorithm are given in Section 3.2. Section 3.3 provides some rudimentary properties of our approach and relates it to stochastic MPC. The computational workload, including the propagation of the state particle state filter, and the possibility of parallel computing of the proposed algorithm are explained in Section 3.4. The algorithm's special case, for linear systems with polyhedral constraints, is discussed in Section 3.5, along with a result showing that, in the unconstrained case, the algorithm yields the

conditional mean. These properties of the algorithm in the linear, quadratic, polyhedral case are the strongest technical support for the algorithm. Section 3.6 includes numerical examples for the nonlinear case and linear special case.

3.2 Problem Formulation

At the current time $k = 0$, we possess the following information.

I. Discrete-time state dynamics

$$x_{k+1} = f(x_k, u_k, w_k), \tag{3.1}$$

where state $x_k \in \mathbb{R}^{r_x}$, control input $u_k \in \mathbb{R}^{r_u}$, exogenous disturbance $w_k \in \mathbb{R}^{r_w}$.

II. Stochastic disturbance process, $\{w_k\}$, assumed to be independent and identically distributed (i.i.d.) possessing known density $\mathcal{W}$. State x_0 is independent of w_k for all k.

III. State and input constraint sets, $\mathbb{X}$ and $\mathbb{U}$, respectively, and $\epsilon \in [0, 1)$, an acceptable probabilistic constraint violation rate.

IV. Initial state x_0-density p_0, provided as a collection of particles $\Xi = \{\xi_0^i \in \mathbb{R}^{r_x}, j = 1, \ldots, L\}$.

V. Nominal full-state-feedback control law $u_k = \kappa(x_k)$, which is (so far) neither assumed to be recursively feasible with respect to $\mathbb{X}$ and $\mathbb{U}$ nor optimal with respect to the following, or indeed any, cost.[1]

VI. An N-stage finite-horizon trajectory cost function J

$$J = \sum_{k=0}^{N} \ell_k(x_k, u_k).$$

[1]While the nominal control law need not be feasible nor optimal, its selection is material for the performance and feasibility of state selection. This is demonstrated in Section 3.6 by evaluation of the state selection algorithm with several controllers.

Because the initial state x_0 is not precisely known (unless p_0 is a point mass function), we seek to select a candidate state value, $x_0^\star$, supported by the initial density which yields: feasibility, perhaps probabilistically or statistically in simulation, on the finite horizon; and, a favorable influence on the subsequent evaluation of the trajectory cost J in ensemble average over the particles.

If the dynamics in (3.1) have $w_k s$ zero and x_0 known, we refer to this as the *deterministic* case.

Next we propose an algorithm, *the State Selection Algorithm*, which returns a *candidate state* $x_0^\star$, which is fed to the control $u_0 = \kappa(x_0^\star)$ and applied to the system. The system output is used to update the particle filter density and the process repeated with new current time $k = 0$. This will be shown by the examples provided in Section 3.6.

State Selection Algorithm

1. Select sample repetition number M and statistical feasibility tolerance $\alpha \in [0, \epsilon)$, parameters of the algorithm.

2. For each $i \in \{1, 2, \ldots, L\}$, choose $x_0' = \xi^i \in \Xi$.

   (a) With (Nr_w)-vectors

$$W_j' = \begin{pmatrix} w_{0,j}'^T & \ldots & w_{N-1,j}'^T \end{pmatrix}^T,$$
$$W_j'' = \begin{pmatrix} w_{0,j}''^T & \ldots & w_{N-1,j}''^T \end{pmatrix}^T,$$

   sample, from $\mathcal{W}^N$ and Ξ, M independent realizations of the $(2Nr_w + r_x)$-vector sequence

$$\left\{ \begin{pmatrix} W_j' \\ W_j'' \\ x_{0,j}'' \end{pmatrix} : j = 1, \ldots, M \right\}.$$

(b) For each j and k, compute the κ-closed-loop state sequence from x'_0, that is, $x'_{0,j} = x'_0$.

$$x'_{k+1,j} = f(x'_{k,j}, \kappa(x'_{k,j}), w'_{k,j}). \tag{3.2}$$

This defines the M samples of the N-long closed-loop control sequence $\{\kappa(x'_{k,j})\}_{k=0}^{N-1}$ from the current initial state candidate x'_0.

(c) For each $j \in \{1, \ldots, M\}$ and $k \in \{0, N-1\}$, compute the open-loop-controlled state sequence from $x''_{0,j}$,

$$x''_{k+1,j} = f(x''_{k,j}, \kappa(x'_{k,j}), w''_{k,j}). \tag{3.3}$$

(d) Compute for each k the sample-average closed-loop control violation rate,[2]

$$\hat{\beta}_k(x'_0) = \frac{1}{M} \sum_{j=1}^{M} \mathbb{1}(\kappa(x'_{k,j}) \in \mathbb{U}). \tag{3.4}$$

(e) Compute for each k the sample-average open-loop-controlled state violation rate,

$$\hat{\lambda}_k(x'_0) = \frac{1}{M} \sum_{j=1}^{M} \mathbb{1}(x''_{k,j} \in \mathbb{X}). \tag{3.5}$$

(f) If $\hat{\beta}_k(x'_0) \geq 1 - \alpha$ and $\hat{\lambda}_k(x'_0) \geq 1 - \alpha$, for all k, then declare this candidate state, x'_0, to be *feasible* and proceed. Otherwise, return to Step 2.

(g) If x'_0 is feasible, calculate its sample-average performance,

$$J_c^M(x'_0) = \frac{1}{M} \sum_{j=1}^{M} \sum_{k=0}^{N} \ell_k(x''_{k,j}, \kappa(x'_{k,j})).$$

[2] $\mathbb{1}(\cdot)$ is the indicator function of an event.

3. Pick $x_0^\star$ to be the feasible x_0' minimizing $J_c^M(\cdot)$, provided the feasible set is non-empty.

In this algorithm, the sequences $\{x'_{k,j}\}_k$ and $\{x''_{k,j}\}_k$ are realizations of the stochastic processes generated by recursions (3.2) and (3.3) driven by constructed independent white noise sequences W_j' and W_j''.

The processes are functions of κ and x_0'. To keep our notation compact, we omit this dependency. For a large number of realizations, M, the average cost $J_c^M(\cdot)$ converges, under regularity conditions [43, 13],

$$\begin{aligned} J_c(x_0') &= \mathbb{E}\left(\sum_{k=0}^{N} \ell_k\left(x_k'', \kappa\left(x_k'\right)\right)\right), \\ &= \mathbb{E}_{x_0''}\mathbb{E}_{W'}\mathbb{E}_{W''}\left(\sum_{k=0}^{N} \ell_k\left(x_k'', \kappa\left(x_k'\right)\right)\right). \end{aligned} \tag{3.6}$$

The expectation is factored due to the mutual independence between the elements x_0'', W', W'' [43]. Further, the sample averages of indicator functions in (3.4) and (3.5) converge to probabilities of constraint satisfaction.

We shall denote the set of feasible state values x_0' under this probability distribution as

$$\begin{aligned} \mathbb{X}_0^\epsilon = \{x_0' \in \Xi |\, &\mathbb{P}(x_k'' \in \mathbb{X}) \geq 1-\epsilon,\ k = 0, 1, \ldots, N, \\ &\mathbb{P}(\kappa(x_k') \in \mathbb{U}) \geq 1-\epsilon,\ k = 0, 1, \ldots, N-1\}. \end{aligned} \tag{3.7}$$

We define the *candidate state* $x_0^\star$ as

$$x_0^\star = \arg\min_{x_0' \in \mathbb{X}_0^\epsilon} J_c(x_0'). \tag{3.8}$$

The corresponding cost of $x_0^\star$ is $J_c^\star$

$$J_c^\star = \min_{x_0' \in \mathbb{X}_0^\epsilon} J_c(x_0') = J_c(x_0^\star). \tag{3.9}$$

Although these definitions are predicated on an infinite M, it will be shown later that M of $\mathcal{O}(\log L)$ is sufficient for providing feasibility and optimality guarantees with probability/reliability at least $1 - \delta$. For example, $M = 135$ is used in the numerical examples in Section 3.6. Where proofs are developed in the next two sections, the results are derived in terms of expectations and probabilities.

An interpretation of the central recursions (3.2) and (3.3) is that, for each particle $x'_0 \in \Xi$, (3.2) generates M control sequences $\{u_{k,j} = \kappa(x'_{k,j})\}$ of a stochastically excited closed-loop system. Recursion (3.3) then uses sample averages of the open-loop cost of these closed-loop sequences themselves then averages over the particle state density. In a sense, the search is over the performance, averaged along the horizon N and over the particle density Ξ, of those randomized closed-loop sequences as functions of x'_0.

Regularity conditions for the State Selection Algorithm

Step (2b) above creates a collection of M closed-loop control sequences $\{\kappa(x'_{k,j})\}_{k=0}^{N-1}$, which are functions of $x'_{0,j}$ and the $w'_{k,j}$s; the steps after that examine the feasibility with respect to $(\mathbb{X}, \mathbb{U})$ and performance with J. In order that this stage is informative for state selection, it is important that the states and control law are suitable excited by differing x'_0 and by w'_k. Considering the system

$$x'_{k+1} = f(x'_k, \kappa(x'_k), w'_k), \quad x'_0, \tag{3.10}$$

$$u'_k = \kappa(x'_k), \tag{3.11}$$

the algorithm requires the accessibility of (3.10) from process noise w'_k, and the observability and reconstructibility of the pair (3.10)-(3.11). This is to ensure the sensitivity of the control sequences to the initial state value and the noise process. These are conditions on the system and control law κ. Without a diverse set of control sequences for selecting x'_0, the non-emptiness of $\mathbb{X}_0^\epsilon$ cannot be assured and the minimization be effective.

As with stochastic optimal control, the closed-loop performance rests on both the control law, κ, and on the information state, Ξ. The propagation of the particle filter should avoid depletion issues and should properly reflect the conditional state density. We assume that it contains the conditional mean, for example, and is sufficiently encompassing to yield a rich set of feasible control sequences.

Other than these remarks, we do not delve deeper. Although, in the computed examples we point to problems with accessibility of feasible states and with the performance limitations when the particle filter is too localized.

3.3 Properties of the candidate state

Borrowing from certainty equivalence control, where the least-squares-best state estimate, $\hat{\xi}_{0|0}$ the conditional mean of Ξ, is selected as the candidate state, we compare the cost J_c for $x_0^\star$ versus that for $\hat{\xi}_{0|0}$.

Proposition 3. *Suppose that $\hat{\xi}_{0|0}$, the sample average of the particles in Ξ, is feasible, that is, $\hat{\xi}_{0|0} \in \mathbb{X}_0^\epsilon$. Then $J_c(x_0^\star) \;=\; J_c^\star \leq J_c(\hat{\xi}_{0|0})$.* □

This is an immediate consequence of the minimization over x_0' in the final stage of the state selection algorithm.

Next, we define and compare the design cost function J_{des} for comparison purposes with the cost J_c.

Definition (design cost). *Assuming full state feedback in (3.1), the design cost of state value x_0' and its associated optimal causal control law $\kappa^\star$ are given by*

$$\begin{aligned} J_{des}(x_0') &= \min_{\kappa} \mathbb{E}_{W'} \left(\sum_{k=0}^{N} \ell_k(x_k', \kappa(x_k')) \right), \\ &= \mathbb{E}_{W'} \left(\sum_{k=0}^{N} \ell_k(x_k', \kappa^*(x_k')) \right), \end{aligned} \tag{3.12}$$

where κ^* is the minimizing causal control law. In the first line of (3.12), $\{x'_k\}_{k\geq 0}$ is constructed using nominal control law κ, while in the second line, using $\kappa^\star$. The minimizer is assumed to exist, otherwise $\min$ is replaced by $\inf$.

Proposition 4. *If the initial state is known and feasible, that is, p_0 is a point mass located at $z \in \mathbb{R}^{r_x}$, and $\mathbb{X}_0^\epsilon = \{z\}$, then $J_{des}(z) \leq J_c^*$.*

Proof.

$$\begin{aligned}
J_c^* &= J_c(z), \\
&= \mathbb{E}_{W'}\mathbb{E}_{W''}\left(\sum_{k=0}^{N} \ell_k\left(x''_k, \kappa^*\left(x'_k\right)\right)\right), \\
&\geq \mathbb{E}_{W'}\mathbb{E}_{W''}\left(\sum_{k=0}^{N} \ell_k\left(x'_k, \kappa^*\left(x'_k\right)\right)\right), \\
&= \mathbb{E}_{W'}\left(\sum_{k=0}^{N} \ell_k\left(x'_k, \kappa^*\left(x'_k\right)\right)\right), \\
&= J_{des}(z),
\end{aligned}$$

where the inequality is due to $\{x'_k\}_k$ being the optimal sequence resulting from applying the optimal control law κ^* and starting from $x'_0 = x''_0 = z$. □

Notice that for J_{des} full state feedback is assumed for all $k \geq 0$, while the hypothesis of Proposition 4 assumes full state feedback at time-0 only, in J_c^*.

Proposition 5. *In the deterministic case, that is, $\Xi = \{z\}$ and $w_k = 0$, for all k, if z is feasible, then $J_{des}^*(z) = J_c^*$.*

Proof. Having $W' = W'' = 0$ and $x'_0 = x''_0 = z$ imply $x'_k = x''_k$ for all k, by the definitions (3.2)

and (3.3) of these two state sequences. Thus,

$$\begin{aligned} J_c^* &= J_c(z), \\ &= \mathbb{E}_{W'}\mathbb{E}_{W''}\left(\sum_{k=0}^{T}\ell_k\left(x_k'', \kappa^*\left(x_k'\right)\right)\right), \\ &= \left(\sum_{k=0}^{N}\ell_k\left(x_k', \kappa^*\left(x_k'\right)\right)\right), \\ &= J_{des}(z), \end{aligned}$$

□

Proposition 5 is a restatement that for the *deterministic case*, open- and closed-loop controls are equivalent and full state feedback for $k \geq 0$ is equivalent to full state feedback at $k = 0$ only. In such case, the *candidate state* for the optimal control κ^* is the true state itself.

3.4 Computational complexity of the state selection algorithm

First, a lower bound on M is found, for guaranteeing ϵ-probabilistic feasibility with a margin δ. Then, the computation time required for applying *the state selection algorithm* is discussed.

3.4.1 $M = \mathcal{O}(\log L)$

From Section 3.2, the initial state density is given as a finite particle mass function over the particle set $\Xi = \{\xi^i | \, i = 1, 2, \ldots, L\}$.

$$p_0(\cdot) = \sum_{i=1}^{L}\delta(\cdot - \xi^i), \tag{3.13}$$

where δ is the Dirac-delta function on $\mathbb{R}^{r_x}$. Since Ξ is finite, so too is $\mathbb{X}_0^\epsilon$ in (3.7), can be written

$$\mathbb{X}_0^\epsilon = \bigcap_{k=1}^{N} \{x_0' \in \Xi |\, \mathbb{P}(x_k'' \in \mathbb{X}) \geq 1 - \epsilon\} \bigcap \bigcap_{k=0}^{N-1} \{x_0' \in \Xi |\, \mathbb{P}(\kappa(x_k') \in \mathbb{U}) \geq 1 - \epsilon\}.$$

By construction of the random vectors x_k'' and x_k', the condition $\{x_k'' \in \mathbb{X}\}$ depends on x_0', x_0'', W' and W'', while $\{\kappa(x_k') \in \mathbb{U}\}$ depends on x_0' and W'. Hence, both are functions of x_0' and $\Lambda = (x_0'', W', W'')$. They can be rewritten as

$$G_k(x_0', \Lambda) = \{x_k'' \in \mathbb{X}\},\ k = 1, \ldots, N,$$

and

$$G_{k+N+1}(x_0', \Lambda) = \{\kappa(x_k') \in \mathbb{U}\},\ k = 0, \ldots, N-1.$$

Hence, more compactly,

$$\mathbb{X}_0^\epsilon = \bigcap_{k=1}^{2N} \{x_0' \in \Xi |\, \mathbb{P}(G_k(x_0', \Lambda)) \geq 1 - \epsilon\}.$$

Statistical feasibility tolerance, α, at Step 1 of the algorithm satisfies $\alpha \in [0, \epsilon)$ and Λ^j, for $j = 1, 2, \ldots, M$, at Step 2a, are independent Monte Carlo samples of Λ. Define the set

$$\mathbb{X}_0^{\alpha,M} = \bigcap_{k=1}^{2N} \{x_0' \in \Xi |\, \frac{1}{M} \sum_{j=1}^{M} \mathbb{1}(G_k(x_0', \Lambda^j)) \geq 1 - \alpha\}. \tag{3.14}$$

This set is a subset of $\mathbb{X}_0^\epsilon$ with probability dependent on the number of samples, M. For a given reliability $\delta \in (0, 1)$, a sufficiently large value of M can be found so that the computed sample average, $\mathbb{X}_0^{\alpha,M}$, provides a suitable approximation of $\mathbb{X}_0^\epsilon$.

The following theorem is a version of Theorem 5 of [30] modified to suit the problem formulation of this chapter.

Theorem 4. *For any $\delta \in (0,1)$, if*

$$M \geq \frac{1}{2(\epsilon-\alpha)^2} \log\left(\frac{L}{\delta}\right), \tag{3.15}$$

then

$$\mathbb{P}(\mathbb{X}_0^{\alpha,M} \subseteq \mathbb{X}_0^{\epsilon}) \geq 1-\delta. \tag{3.16}$$

Lemma 2. ***(Hoeffding's inequality [17]).*** *For independent random variables Z_q, $q = 1,\ldots,\bar{M}$, $\mathbb{P}(Z_q \in [a_q, b_q]) = 1$, $a_q \leq b_q$, for all $t \geq 0$*

$$\mathbb{P}\left(\sum_{q=1}^{\bar{M}} (Z_q - \mathbb{E}\ Z_q) \geq t\bar{M}\right) \leq \exp\left(-\frac{2\bar{M}^2t^2}{\sum_{q=1}^{\bar{M}}(b_q-a_q)^2}\right)$$

□

Proof. (of Theorem 4). Let $x \in \Xi\backslash\mathbb{X}_0^{\epsilon}$. Then there exists l, a minimizer of $\mathbb{P}(G_l(x,\Lambda))$, $l \in \{1,2,\ldots,N\}$. By definition of x, we have $\mathbb{P}(G_l(x,\Lambda)) < 1-\epsilon$. Define the random variables Y_j, for $j = 1,2,\ldots,M$, such that $Y_j = \mathbb{1}\{G_l(x,\Lambda^j)\}$. Consequently,

$$\mathbb{E}\, Y_j = \mathbb{P}\left(G_l(x,\Lambda^j)\right) < 1-\epsilon. \tag{3.17}$$

Hence,

$$
\begin{aligned}
&\mathbb{P}(x \in \mathbb{X}_0^{\alpha,M}) \\
&\leq^{①} \mathbb{P}\left(x \in \{x_0' \in \Xi | \, \frac{1}{M}\sum_{j=1}^{M} \mathbb{1}(G_l(x_0', \Lambda^j)) \geq 1-\alpha\}\right), \\
&=^{②} \mathbb{P}\left(\frac{1}{M}\sum_{j=1}^{M} Y_j \geq 1-\alpha\right), \\
&\leq^{③} \mathbb{P}\left(\frac{1}{M}\sum_{j=1}^{M} (Y_j - \mathbb{E}\, Y_j) \geq -1+\epsilon+1-\alpha\right), \\
&\leq \mathbb{P}\left(\sum_{j=1}^{M} (Y_j - \mathbb{E}\, Y_j) \geq M(\epsilon-\alpha)\right), \\
&\leq^{④} \exp\left(-2M(\epsilon-\alpha)^2\right),
\end{aligned}
$$

where the numbered inequalities follow from: ①- probability of an intersection of events underbounds that of any single event; ②, ③- the definition of Y_j in (3.17); ④- Hoeffding's inequality, Lemma 2, with $a_j = 0,\ b_j = 1$ for all $j \in \{1, 2, \ldots, M\}$. Therefore,

$$
\begin{aligned}
\mathbb{P}\left(\mathbb{X}_0^{\alpha,M} \not\subseteq \mathbb{X}_0^{\epsilon}\right) &= \mathbb{P}\left\{\text{there exists } x \in \mathbb{X}_0^{\alpha,M} \textit{ s.t. } x \notin \mathbb{X}_0^{\epsilon}\right\}, \\
&\leq \sum_{x \in \Xi \setminus \mathbb{X}_0^{\epsilon}} \mathbb{P}(x \in \mathbb{X}_0^{\alpha,M}), \\
&\leq L \exp\left(-2M(\epsilon-\alpha)^2\right).
\end{aligned}
$$

If the left-hand-side is to be $\delta \in (0,1)$, then, taking the $\log(\cdot)$ of both sides, results in (3.15). □

If $x_0' \in \mathbb{X}_0^{\alpha,M}$, then $x \in \mathbb{X}_0^{\epsilon}$ with probability at least $1-\delta$. A similar result providing a lower bound of M of $\mathcal{O}(\log L)$ can be found for achieving optimality: x_0' in Step 3 of the algorithm satisfies (3.8) with probability close to one, using an extension of Corollary 6 of [30].

The approximations inherent in selecting parameters $\alpha \in (0, \epsilon)$ and $\delta \in (0,1)$, together with the imprecision in the inequality (3.15), are the source of the inbuilt conservatism of the

quantifications of the State Selection Algorithm. This will become evident in the achieved constraint violation rates in Section 3.6 where computational examples are conducted with sample repetition value $M = 135$ and particle count $L = 400$.

3.4.2 Computation time

The State Selection Algorithm has computational complexity $\mathcal{O}(NL \log L)$, per time-step. It is parallelizable in: its second step; across i in the choice of x_0'; and also across j over the samples. Using a graphics processing unit (GPU) would further decrease the computation time required.

As the dimension, r_x, of the state increases, the complexity of this approach increases via the requisite (perhaps dramatic) increase in the number of samples, L, of the particle filter. Once L is fixed, however, the calculations for M given ϵ, α and δ abide.

3.5 Constrained stochastic linear systems with quadratic cost

In this section, we show that under the assumptions of: linear dynamics, linear state feedback control law, quadratic stage costs, and polyhedral state and input constraints, the candidate state is the solution of a quadratic program over $\mathbb{R}^{r_x}$ based on the second-order moments of the densities. This obviates the need for: calculation of M sample sequences and their sample averages of constraint violations and costs, and the inclusion of margin parameters α and δ earlier. We assume that the filtered conditional state density is described by its first two moments, which, in turn, might be provided by a Kalman filter or particle filter.

At time $k = 0$, we have:

I. Discrete-time linear state dynamics

$$x_{k+1} = Fx_k + Gu_k + w_k. \tag{3.18}$$

II. $\{w_k\}$ is i.i.d., $\mathbb{E}\, w_k = 0$ and $\text{cov}(w_k) = \Sigma_w$, for all k.

III. Minimal constraint violation probability $\epsilon \in (0, 1)$ for polyhedral state and input constraint sets,

$$\begin{aligned} \mathbb{X} &= \{x_k \in \mathbb{R}^{r_x} | Tx_k \leq \bar{x}\},\ \bar{x} \in \mathbb{R}^t, \\ \mathbb{U} &= \{u_k \in \mathbb{R}^{r_u} | Su_k \leq \bar{u}\},\ \bar{u} \in \mathbb{R}^m, \end{aligned} \tag{3.19}$$

where $T \in \mathbb{R}^{t \times r_x}$ and $S \in \mathbb{R}^{m \times r_u}$ have full row rank.

IV. The first two moments, $\mathbb{E}\, x_0 = \hat{x}_0$ and $\text{cov}(x_0) = \Sigma_0$, of the density of the initial state x_0, which is independent from w_k for all k.

V. Linear full-state-feedback control law $u_k = Kx_k$, which is yet neither assumed to be recursively feasible with respect to $\mathbb{X}$ and $\mathbb{U}$ nor optimal with respect to a cost, nor indeed stabilizing.

VI. A finite-horizon quadratic trajectory cost function J

$$J = x_k^T Q_N x_k + \sum_{k=0}^{N-1} \left[x_k^T Q x_k + u_k^T R u_k \right].$$

We derive a variant of the State Selection Algorithm for this case, where expectations of states, costs, constraint violation are directly characterized without the need for sample averages of simulations. The net result is a quadratic program to determine the selected state $x_0^\star$. This is tantamount to operating the State Selection Algorithm with very large M and without the attendant computational demands.

We shall see in the numerical examples later that the state selection algorithm for these constrained linear quadratic problems returns the conditional mean when the constraints are inactive.

The state sequences in (3.2) and (3.3) become

$$x'_{k+1} = (F + GK)x'_k + w'_k,\ x'_0 \in \mathbb{R}^{r_x}, \tag{3.20}$$

$$x''_{k+1} = Fx''_k + GKx'_k + w''_k,\ x''_0 \sim p_0. \tag{3.21}$$

These recursions from the State Selection Algorithm yield:

- x'_k affine in x'_0 and w'_{k-j},
- x''_k affine in x'_0, x''_0, w'_{k-j} and w''_{k-j}.

In turn, this implies that the expected cost function, J_c of (3.6), is quadratic in x'_0. The analysis of the constraints is more work but results in new, but more conservative, polyhedral constraints on x'_0.

3.5.1 The cost function J_c

For $x'_0 \in \mathbb{X}_0^\epsilon$, the cost function J_c is given by (3.6).

$$J_c(x'_0) = \mathbb{E}_{x''_0}\mathbb{E}_{W'}\mathbb{E}_{W''}\left(\sum_{k=0}^{N-1}\left[(x''_k)^T Q x''_k + (x'_k)^T K^T R K x'_k\right] + (x''_N)^T Q_N x''_N\right), \tag{3.22}$$

$$= \mathbb{E}_{x''_0}\mathbb{E}_{W'}\mathbb{E}_{W''}\left(\sum_{k=0}^{N-1}\left[\mathit{tr}(Qx''_k(x''_k)^T) + \mathit{tr}(K^T R K x'_k(x'_k)^T)\right] + \mathit{tr}(Q_N x''_N(x''_N)^T)\right). \tag{3.23}$$

From (3.20) and (3.21),

$$x'_k = F_K^k x'_0 + \sum_{p=0}^{k-1} F_K^p w'_{k-p-1}, \tag{3.24}$$

$$x''_k = F^k x''_0 + \sum_{j=0}^{k-1} F^j w''_{k-1-j} + \Psi_{k-1}x'_0 + \sum_{h=0}^{k-1} F^h GK \sum_{p=0}^{k-h-2} F_K^p w'_{k-h-2-p}, \tag{3.25}$$

where $F_K = (F + GK)$ and $\Psi_{k-1} = \sum_{h=0}^{k-1} F^h GK F_K^{k-h-1}$. That is, these state sequences are affine functions of x_0'' and x_0'. Hence, the cost J_c is quadratic in x_0' and x_0''.

Denoting mean values, $\hat{x}_k' = \mathbb{E}_{W'} x_k'$ and $\hat{x}_k'' = \mathbb{E}_{x_0''} \mathbb{E}_{W'} \mathbb{E}_{W''} x_k''$, we have

$$\hat{x}_k' = F_K^k x_0', \quad \hat{x}_k'' = F^k \hat{x}_0'' + \Psi_{k-1} x_0'.$$

Hence, the errors $\tilde{x}_k' = x_k' - \hat{x}_k'$ and $\tilde{x}_k'' = x_k'' - \hat{x}_k''$ are zero by dint of w_k' and w_k'' being zero mean.

Using independence assumptions, and ignoring additive constants,

$$J_c(x_0') = (x_0')^T \mathcal{A}_1 x_0' + (\hat{x}_0'')^T \mathcal{A}_2 x_0'. \tag{3.26}$$

The formulæ for and derivation of $\mathcal{A}_1$ and $\mathcal{A}_2$ can be found in the Appendix.

Corollary 2. *Suppose that K_k is the optimal time-variant feedback gain of the unconstrained N-horizon LQ problem with Q, R and Q_N. Then the candidate state is the conditional mean.* □

This result follows directly from the discrete-time equivalent of the argument provided in [1, p. 221] and the monotonic dependence of the control performance on the trace of the state estimate covariance in the unconstrained case.

3.5.2 The constraints

The state and input sequences $\{x_k''\}$, $\{u_k\} = \{Kx_k'\}$ are both stochastic by construction. The prediction covariance, Σ_k', of x_k' can be described by

$$\Sigma_{k+1}' = (F + GK)\Sigma_k'(F + GK) + \Sigma_w,\ \Sigma_0' = 0, \tag{3.27}$$

where the initial covariance is zero because x'_0 is deterministic. For Σ''_k, the covariance of x''_k, we have

$$\Sigma''_{k+1} = F\Sigma''_k F^T + GK\Sigma'_k K^T G^T + \Sigma_w,\ \Sigma''_0 = \Sigma_0. \tag{3.28}$$

The initial covariance Σ_0 is provided at Item IV. in the linear problem statement.

The input sequence covariance follows from (3.27).

$$\text{cov}(u_k) = \text{cov}(Kx'_k) = K\Sigma'_k K^T. \tag{3.29}$$

Next, we show how the probabilistic constraints in (3.19) can be transformed into deterministic linear constraints in terms of the state sequence means (3.26) and covariances (3.27), (3.28).

Proposition 6. *The probabilistic polyhedral state constraints in (3.19) are satisfied if the following deterministic polyhedral constraints are satisfied*

$$T\hat{x}''_k \leq \bar{x} - \sqrt{\frac{t-\epsilon}{\epsilon}}\sqrt{diag(T\Sigma_k T^T)}, \tag{3.30}$$

where: t is the number of rows of T_k, the function diag$(\cdot)$ returns the diagonal of a square matrix as a column vector and the second square root is elementwise.

Lemma 3. ***(Cantelli's inequality)*** *For a scalar random variable γ with mean $\hat{\gamma}$ and variance Γ,*

$$\mathbb{P}(\gamma - \hat{\gamma} \geq \eta) \leq \frac{\Gamma}{\Gamma + \eta^2},\ \eta \geq 0. \tag{3.31}$$

□

Lemma 4. *For $j = 1, \ldots, t$, let $T(j)$ be the j^{th} row of T and $\bar{x}(j)$ be the j^{th} element of $\bar{x}$. The*

probabilistic constraints

$$\mathbb{P}\Big(T(j)x''_k \leq \bar{x}(j)\Big) \leq 1 - \frac{\epsilon}{t}, \tag{3.32}$$

are satisfied if the following linear inequality holds

$$T(j)\hat{x}''_k \leq \bar{x}(j) - \sqrt{\frac{t-\epsilon}{\epsilon}}\sqrt{T(j)\Sigma''_k T(j)^T}. \tag{3.33}$$

Proof. Analogous to the work in [14], suppose there exists $\rho \geq 0$ such that

$$T(j)\hat{x}'' \leq \bar{x} - \rho, \tag{3.34}$$

hence the condition $T(j)x'' \geq T(j)\hat{x}'' + \rho$ is implied by $T(j)x'' \geq \bar{x}(j)$, so that,

$$\begin{aligned}
\mathbb{P}\Big(T(j)x'' \geq \bar{x}\Big) &\leq \mathbb{P}\Big(T(j)x'' \geq T(j)\hat{x}'' + \rho\Big), \\
&= \mathbb{P}\Big(T(j)x'' - T(j)\hat{x}'' \geq \rho\Big), \\
&\leq \frac{T(j)\Sigma'' T(j)^T}{T(j)\Sigma'' T(j)^T + \rho^2},
\end{aligned}$$

where the last inequality follows from Cantelli's inequality (3.31). If ρ is sufficiently large that the last term is upper bounded by ϵ/t,

$$\frac{T(j)\Sigma'' T(j)^T}{T(j)\Sigma'' T(j)^T + \rho^2} \leq \frac{\epsilon}{t},$$

or equivalently

$$\rho \geq \sqrt{\frac{t-\epsilon}{\epsilon}}\sqrt{T(j)\Sigma'' T(j)^T}.$$

This lower bound of ρ, when used in (3.34), yields (3.33). □

Lemma 5. *Let $(\Omega, \mathcal{B}, \mathbb{P})$ be a probability space and $E_i \in \mathcal{B}$ for $i = 1, \ldots, n$. If $\mathbb{P}(E_i) \geq 1 - \epsilon/n$, for all $i = 1, \ldots, n$, then $\mathbb{P}(\bigcap_{i=1}^{n} E_i) \geq 1 - \epsilon$.* □

Proof. **(of Proposition 6)**. Notice that the state constraint sets in (3.19) can be written as intersection of sets

$$\begin{aligned}\mathbb{X}_k &= \{x_k \in \mathbb{R}^{r_x} | Tx_k \leq \bar{x}_k\}, \\ &= \bigcap_{j=1}^{t} \{x_k \in \mathbb{R}^{r_x} | T(j)x_k \leq \bar{x}_k(j)\},\end{aligned}$$

since all rows are to be enforced simultaneously [30]. By Lemma 5, $\mathbb{P}(x_k \in \mathbb{X}_k) \geq 1 - \epsilon$ is implied by $\mathbb{P}(\{x_k \in \mathbb{R}^{r_x} | T(j)x_k \leq \bar{x}_k(j)\}) \geq 1 - \epsilon/t$. The latter is implied by (3.33) in Lemma 4. Stacking the inequalities in (3.33) for all of the t rows of T, we get (3.30). Notice that with an increase in the number of rows t, the constraints become tighter and the approximation more conservative. □

Using parallel arguments, results analogous to those above hold for the probabilistic input constraints. Finally, the probabilistic constraints (3.19) remain polyhedral in x'_0 as follows with K being the nominal state feedback gain and S the matrix in the control constraint (3.19).

$$\begin{aligned} T\hat{x}''_k &\leq \bar{x} - \sqrt{\frac{t-\epsilon}{\epsilon}}\sqrt{\mathit{diag}(T\Sigma''_k T^T)}, \\ &\quad \textit{for all } k = 1, 2, \ldots, N, \\ SK\hat{x}'_k &\leq \bar{u} - \sqrt{\frac{m-\epsilon}{\epsilon}}\sqrt{\mathit{diag}(SK\Sigma'_k K^T S^T)}, \\ &\quad \textit{for all } k = 1, 2, \ldots, N-1, \\ SK\hat{x}'_0 &\leq \bar{u}, \end{aligned} \tag{3.35}$$

where $\hat{x}'_k$ and $\hat{x}''_k$ are solely functions of x'_0, as in (3.26). The covariance matrices Σ'_k and Σ''_k can be computed offline and are independent from x'_0. The last inequality is due to the fact that $\Sigma'_0 = 0$.

We follow [58] in the usage of the 'closed-loop covariance' in (3.35). That is, we replace the covariance matrices Σ'_k and Σ''_k by their one step ahead predictions

$$\Sigma'_1 = \Sigma_w, \quad \Sigma''_1 = F\Sigma''_0 F^T + \Sigma_w, \tag{3.36}$$

where $\Sigma''_0 = \Sigma_0$ is given in (3.1). This accounts for the fact that at the next time step, a new measurement of the system will be available. Thus, this relaxes the constraints and avoids the unbounded growth in k of Σ''_k when F is unstable.

The minimization problem, to find $x_0^\star$, in a compact form is a quadratic program in x'_0.

$$\begin{aligned}
&\min_{x'_0} \quad (x'_0)^T \mathcal{A}_1 x'_0 + (\hat{x}''_0)^T \mathcal{A}_2 x'_0, \\
&\quad \textit{subject to} \\
&T\Psi_{k-1} x'_0 \leq \bar{x} - \sqrt{\frac{t-\epsilon}{\epsilon}} \sqrt{\textit{diag}(T\Sigma''_1 T^T)} - TF^k \hat{x}''_0, \\
&\quad \textit{for all } k = 1, 2, \ldots, N \\
&SKF_K^k x'_0 \leq \bar{u} - \sqrt{\frac{m-\epsilon}{\epsilon}} \sqrt{\textit{diag}(SK\Sigma_w K^T S^T)}, \\
&\quad \textit{for all } k = 1, 2, \ldots, N-1 \\
&SKx'_0 \leq \bar{u},
\end{aligned} \tag{3.37}$$

For linear quadratic stochastic problems with polyhedral constraints, the reduction of the State Selection Algorithm to a quadratic program for x'_0 admits an appreciation of the methodology absent the sampling and the sample-average convergence requirements. In turn, this allows comparison with known solutions from the deterministic case [6] and conditional mean state estimates from Kalman filtering.

3.6 Numerical examples

In this section, we present computational examples in which the State Selection Algorithm of Section 3.4 is applied. We also present an example that belongs to the case of linear systems with quadratic cost and polyhedral constraints; amenable to the quadratic program methods presented in Section 3.5. The computational burden of the algorithm is also evaluated, including the propagation of the bootstrap particle filter.

3.6.1 Nonlinear system

We follow the problem formulation schema from Section 3.2.

I. The state dynamics are described by

$$\begin{aligned} z_{k+1} &= 0.9z_k + 0.2h_k + w_k^1, \\ h_{k+1} &= -0.15z_k + 0.9h_k + 0.05z_k h_k + u_k + w_k^2, \\ y_k &= z_k + v_k. \end{aligned}$$

Here $x_k = \begin{pmatrix} z_k & h_k \end{pmatrix}^T \in \mathbb{R}^2$ is the state vector, u_k the scalar control, y_k the scalar measurement, and $w_k = \begin{pmatrix} w_k^1 & w_k^2 \end{pmatrix}^T \in \mathbb{R}^2$ the process noise, and v_k is the scalar measurement noise.

II. The noises $w_k \sim \mathcal{N}(0_2, 0.3\mathbb{I}_2)$ and $v_k \sim \mathcal{N}(0, 0.3)$[3].

III. The input constraint set is $\mathbb{U} = [-3, 3]$. The state constraint set, $\mathbb{X}$, is the complement of $\mathcal{L}$ in $\mathbb{R}^2$, where $\mathcal{L} = [3, 5] \times [-4, 2] \cup [-2, 5] \times [-7, -4]$. This L-shaped set to be avoided is depicted in the figures below. The constraint violation rate is $\epsilon = 0.3$.

IV. The x_0 state density is provided by a collection of $L = 400$ particles in $\mathbb{R}^2$.

V. We consider two successive nominal controllers:

[3] $\mathcal{N}(\mu, \Sigma)$ denotes a Gaussian density with mean vector μ and covariance matrix Σ.

(a) the stabilizing feedback-linearizing control law $u_k = \kappa_1(x_k) = -0.05 z_k h_k$; and then,

(b) a feasible optimal controller, $\kappa_2(x_k)$, to be detailed shortly.

VI. The running cost is $\ell_k(x_k, u_k) = x_k^T x_k + u_k^2$. The horizon $N = 6$.

For the State Selection Algorithm, choose $\alpha = .1$, $\delta = .01$. Theorem 4 then admits $M = 135$. The following sequence is then conducted starting from Ξ being $L = 400$ particles sampled from $\mathcal{N}((\ 7.5 \quad -7.5\ )^T, .5\mathbb{I}_2)$.

i) The selected state, $x_0^\star$, is applied in the control $u_k = \kappa(x_0^\star)$,

ii) The output y_{k+1} is measured,

iii) A bootstrap particle filter computes an updated set, Ξ, of L filtered particles for x_{k+1},

iv) The state selection is re-performed.

With this iteration, the State Selection Algorithm, its attendant particle filter, and the nominal control law can be evaluated jointly for their control performance and constraint handling. Since the chapter purports to study state estimation for control, this is a critical evaluation.

To visualize the open-loop dynamics better, Figure 3.1 displays the streamlines of the state dynamics with zero input and zero state disturbances.

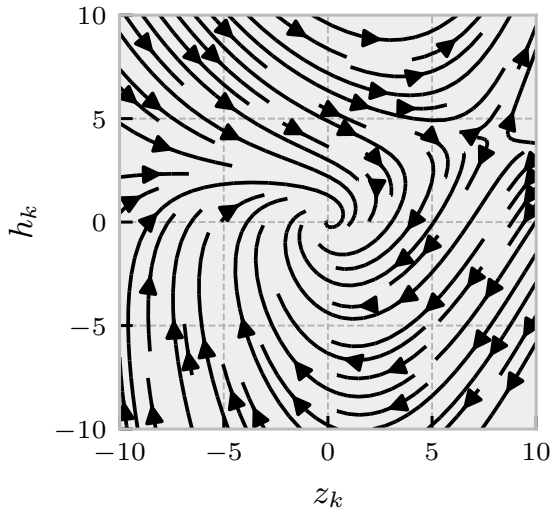

Figure 3.1. The streamlines of the nonlinear system state with u_k and w_k set to zero.

Example 1: stabilizing controller

We conduct two comparative simulations of the controlled nonlinear system: one with the State Selection Algorithm, as outlined above, i.e. $u_k = \kappa_1(x_0^\star)$; and the other with what might be termed the certainty equivalence controller, $u_k = \kappa_1(\hat{x}_{0|0})$ with $\hat{x}_{0|0}$ being the corresponding particle filter conditional mean. In each case the particle filter evolves according to the respective measured output, which in turn depends on the applied control. Figure 3.2 displays the controlled state filtered particle density with the State Selection Algorithm feedback. Figure 3.3 shows the corresponding conditional mean feedback case. Figure 3.4 shows the percentage of the particles, at each time step, which violate the state constraints and land inside the set $\mathcal{L}$. This shows that State Selection Algorithm enforcing more caution.

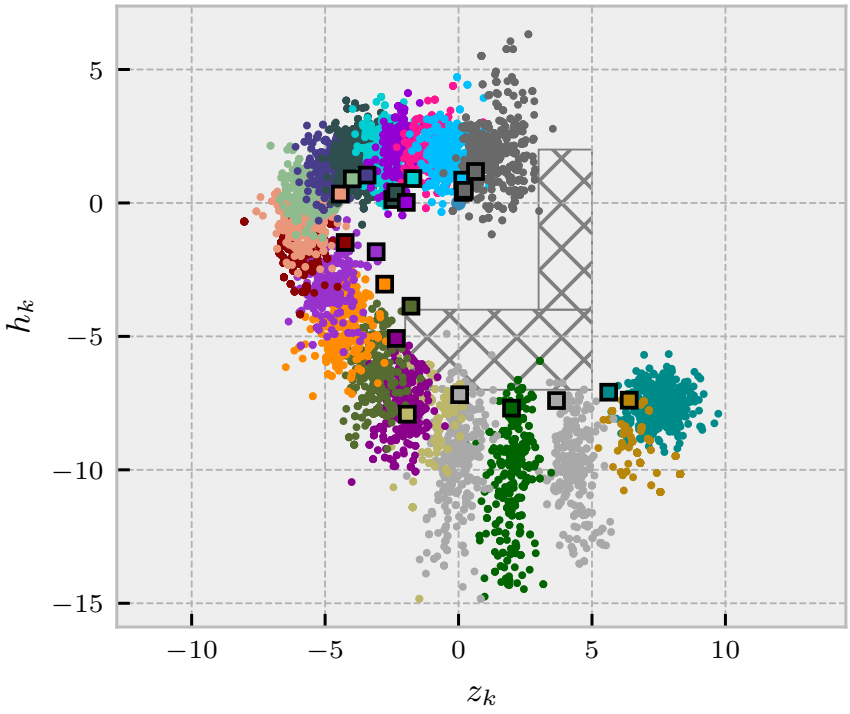

Figure 3.2. State Selection Algorithm: The evolution of the particle filtered density with $u_k = \kappa_1(x_0^\star)$, that is, the *candidate state* $x_0^\star$ being used by the controller at each time step. The black squares indicate the location of the selected states from the particle densities. The cross-hatched object is the complement of the state constraint set, $\mathbb{X}$.

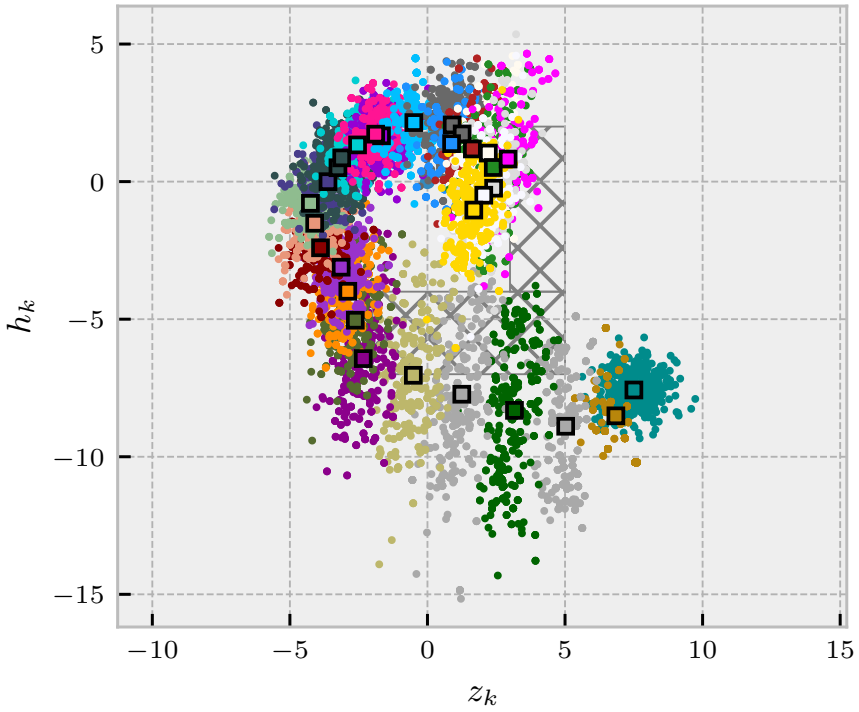

Figure 3.3. Conditional mean: The evolution of the particle filtered density with $u_k = \kappa_1(\hat{x}_{0|0})$, where $\hat{x}_{0|0}$ is the conditional mean of the particle filtered density indicated by the black square at each time.

We make the following observations concerning these two simulations.

- The closed-loop particle densities evolve differently because the control signals, and therefore states and measurements, differ.
- The State Selection Algorithm avoids state constraint violation, perhaps too conservatively, while the conditional mean control violates the state constraint requirements, as measured using the particle density.
- It is evident that the selected state enforces caution into the subsequent particle density by choosing $x_0^\star$ close to the constraint boundary. This is a property dependent on the nominal control law $\kappa_1(\cdot)$.
- The state selection simulation stopped at time-step $k = 22$ because the feasible set of

states in the particle density, $\mathbb{X}_0^{\alpha,M}$, was empty. That is, no state choice x_0' yielded a feasible solution. Again, this is a property stemming from the lack of reachability of the κ_1-controlled dynamic system close to the origin.

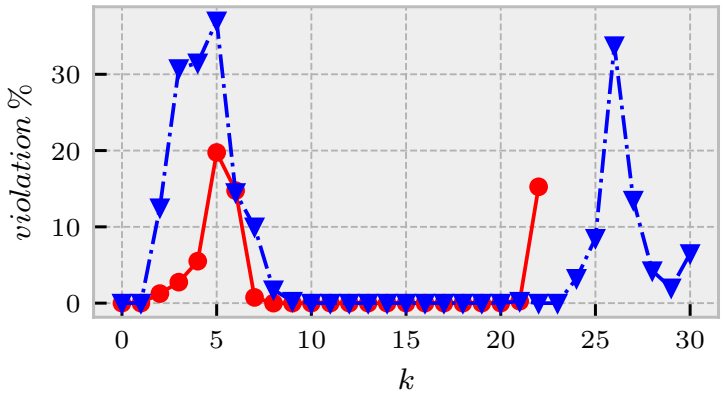

Figure 3.4. The blue triangles indicate the state constraint violation rate for the nonlinear system with control $u_k = \kappa_1(\hat{x}_{0|0})$, that is, using the current conditional mean of the particle density. The red circles indicate the same rate for the control $u_k = \kappa_1(x_0^\star)$ based on the state selection algorithm.

The simulations were conducted using PYTHON, an uncompiled interpretive program, on an M1-chip 2021 MacBook Pro with 16.00 GB of RAM. The average running time for each time step of the simulation in Figure 3.2, comprising the State Selection Algorithm (the dominant load) and the particle filter, is about 4.9 seconds. The algorithm is completely parallelizable, over the state choice x_0' and over the corresponding samples. Although this was not implemented here, it can offer a potential improvement in computation time.

Example 2: feasible optimal controller

We next re-conduct the previous experiment with state feedback controller $\kappa_2(\cdot)$: a $(\mathbb{U}, \mathbb{X})$-feasible, infinite-horizon discounted-cost, optimal controller computed by sampling and a value iteration. Its construction is detailed in the Appendix.

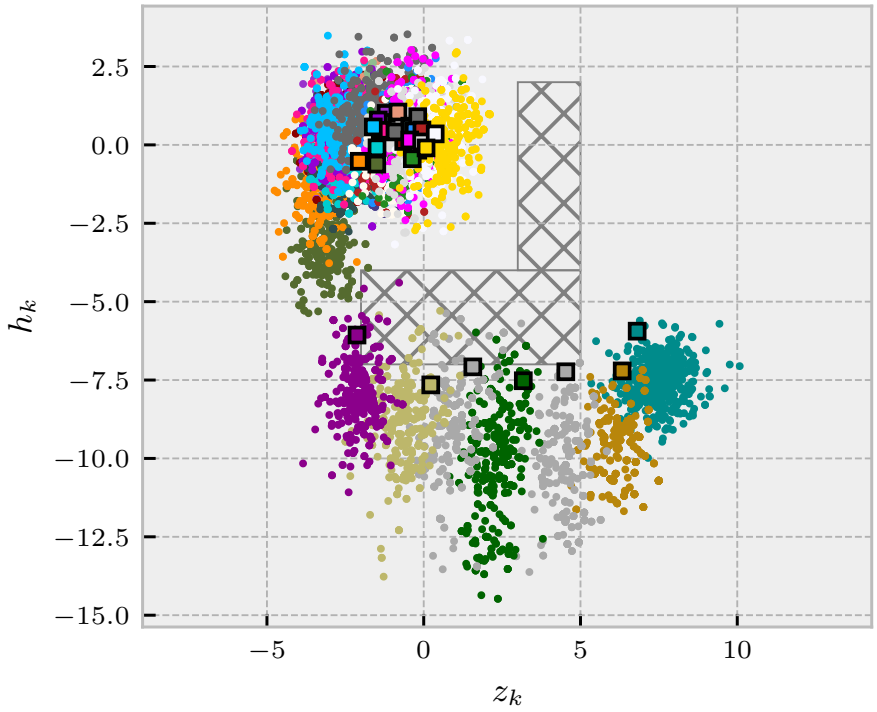

Figure 3.5. State Selection Algorithm: The evolution of the particle filtered density with $u_k = \kappa_2(x_0^\star)$. That is, the *candidate state* $x_0^\star$, indicated by the black squares, being used by the controller at each current time step. The black squares indicate the selected state.

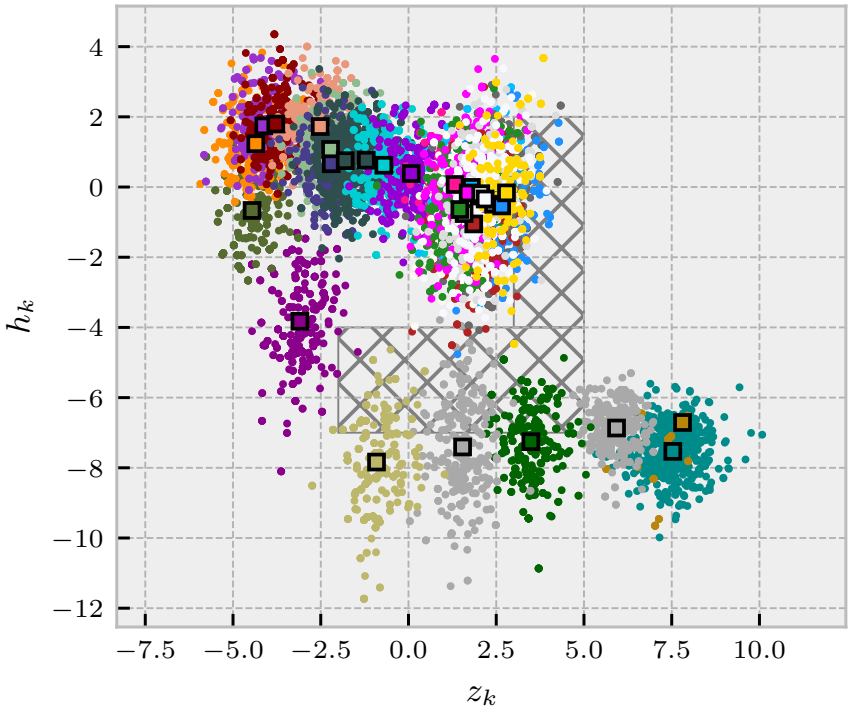

Figure 3.6. Conditional mean: The evolution of the particle filtered density with $u_k = \kappa_2(\hat{x}_{0|0})$, that is, the particle filter conditional mean being used by the controller at each current time step.

For control law $\kappa_2(\cdot)$, two closed-loop simulations were conducted. Figure 3.5 displays the result of using the State Selection Algorithm's $x_0^\star$ in the controller. Figure 3.6 shows the corresponding behavior when the particle filters' conditional mean, $\hat{x}_{0|0}$, is used. Figure 3.7 shows the percentage of the particles, at each time step, which violate the state constraints. The value of ϵ is 0.3.

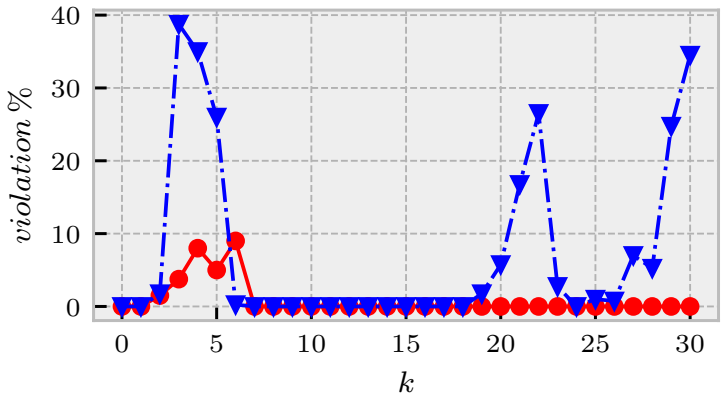

Figure 3.7. The blue triangles indicate the state constraint violation rate for the nonlinear system with control $u_k = \kappa_2(\hat{x}_{0|0})$, that is, using the conditional mean of the particle density. The red circles indicate the same rate for the control $u_k = \kappa_2(x_0^\star)$ based on the state selection algorithm. The value of ϵ is 30%.

Remark. *The information state/particle density Ξ, as discussed in Section 3.2, plays a central role in the State Selection Algorithm; it diversifies the closed-loop control sequences $\{\kappa(x'_{k,j})\}_{k=0}^{N}$, since x'_0 is chosen from the particles in Ξ. Therefore, the requirements of a particle filter in the State Selection Algorithm can be categorized in two parts. Firstly, the particle filter has to be a sufficiently accurate approximation to the Bayesian filter. This is a foundational assumption upon which the algorithm is built. Secondly, particle depletion must be avoided in order that particle variability is preserved.*

In the above examples, we used a measurement disturbance, v_k, of variance 0.3 and state disturbance, w_k, of covariance $0.3\mathbb{I}_2$. This facilitated retaining diversity in the particle filtered density as shown in the figures. Selecting the variance of the measurement disturbance to be 0.1 for instance, with the number of particles fixed to 400, resulted in particle depletion and hence to infeasibility issues with the State Selection Algorithm and furthermore to a poor representation of the Bayesian filter density.

Converse to but alongside the requirements of the particle filter rest those for the control law, κ, and the state disturbance w_k. This subject is broached in Section 3.2 as a regularity condition. For the State Selection Algorithm the accessibility of the state and control signals

from w_k affects the closed-loop feasibility set, $\mathbb{X}_0^\epsilon$, for the algorithm. In the nonlinear example with feedback linearizing controller, κ_1, the feasible set is empty after 22 steps – this changes with each run. This is a result of a diminished control gain near the origin resulting in poor excitation.

Example 3: Linear dynamics with polyhedral constraints

In this example, a standard DC-DC converter regulation problem is considered. It is a benchmark in the stochastic MPC literature and used in [45, 10, 29].

- The dynamics are described by

$$\begin{pmatrix} x_{k+1}^1 \\ x_{k+1}^2 \end{pmatrix} = x_{k+1} = Fx_k + Gu_k + w_k,$$

$$y_k = Hx_k + v_k,$$

where

$$F = \begin{bmatrix} 1 & 0.0075 \\ -0.143 & 0.996 \end{bmatrix}, \quad G = \begin{bmatrix} 4.798 \\ 0.115 \end{bmatrix},$$

$$H = \mathbb{I}_{2\times 2}.$$

- Noise signals w_k and v_k are white with zero means, and independent from each other and from x_0.

$$\text{cov}(w_k) = \Sigma_w = 0.1\mathbb{I}_{2\times 2},\ \text{cov}(v_k) = \text{diag}(0.5, 0.4),$$

$$\mathbb{E}\, x_0 = (0.6455, 1.3751)^T,\ \text{cov}(x_0) = \Sigma_0 = 0.1\mathbb{I}_{2\times 2}.$$

We take the densities to be Gaussian in the simulations.

- The probabilistic constraints are

$$\mathbb{P}(x_k^1 \leq 2) \geq 1 - \epsilon, \text{ for all } k = 1, \ldots, N$$

where $\epsilon = 10\%$, differently from the 40% of [45]. According to Proposition 6 and (3.37), this is implied by

$$T\Psi_{k-1}\hat{x}_0 \leq \bar{x} - \sqrt{\frac{t-\epsilon}{\epsilon}}\sqrt{\text{diag}(T\Sigma_1''T^T)} - TF^k\hat{x}_0'',$$

for all $k = 1, \ldots, N$. Where $T = (1, 0)$, $\bar{x} = 2$, $t = 1$, and Σ_1'' is the prediction covariance and defined in (3.36).

- The linear state-feedback controller $u_k = Kx_k$ is chosen to be the infinite horizon LQR controller with the weighting matrices, similar to those in [45], $Q = \text{diag}(1, 10)$ and $R = 10$

$$K = \begin{bmatrix} -0.2409 & 0.3930 \end{bmatrix},$$

these weighting matrices are also used in forming the optimization problem (3.37), and $Q_N = Q$.

- The prediction horizon, for (3.37), is chosen to be $N = 8$. The associated matrices $\mathcal{A}_1$ and $\mathcal{A}_2$ are

$$\mathcal{A}_1 = \begin{bmatrix} 47.23 & -43.76 \\ -43.76 & 45.51 \end{bmatrix}, \quad \mathcal{A}_2 = \begin{bmatrix} -93.98 & 87.18 \\ 85.33 & -89.45 \end{bmatrix},$$

The quadratic program (3.37) is solved at each time to find the candidate state $x_0^\star$, which is used by the linear state-feedback controller. The state conditional mean and covariance are updated using the Kalman filter, which is the least-squares optimal unbiased estimator still. The result of 100 closed-loop iterations is shown in Figure 3.8. The corresponding simulation

using the state conditional mean from the Kalman filter with the linear controller is shown in Figure 3.9.

Notice that when x_k^2 is below a certain line, the optimization problem returns, approximately, the conditional mean as the candidate state. However, above that line, the candidate state differs. This is the effect of the constraint, present in the state section process but not in LQR or the Kalman filter.

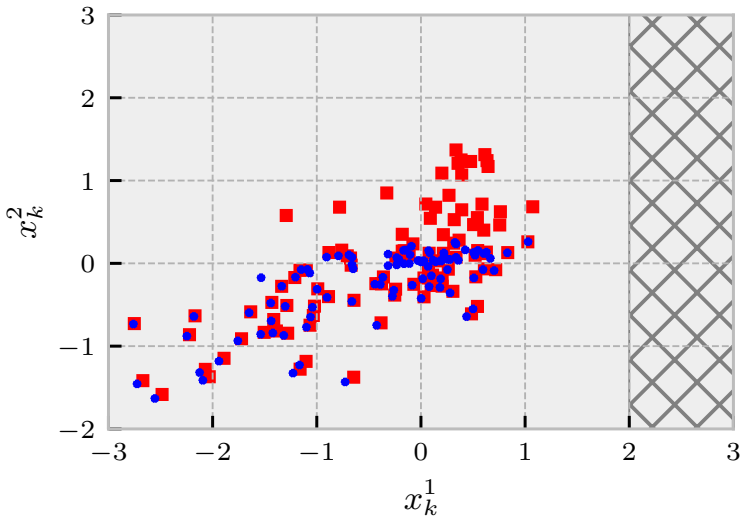

Figure 3.8. Values of the Kalman filter conditional mean (red squares) with control $u_k = Kx_0^\star$. The blue dots indicate the values chosen for the candidate state, $x_0^\star$, used by the controller.

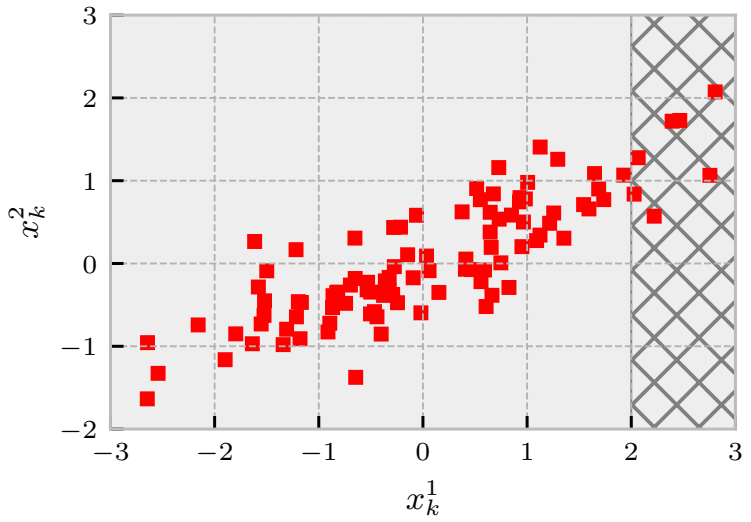

Figure 3.9. Values of the Kalman filter conditional mean, $\hat{x}_{0|0}$, with $u_k = K\hat{x}_{0|0}$.

Figure 3.10 shows: the conditional mean of x_k^1, from Figure 3.8; its two-sigma intervals propagated by the Kalman filter (the square root of the (1,1) entry of the conditional covariance); and the true state. The shaded area is two standard deviations, or 95% confidence interval. Since the tolerance used in the algorithm for this example is $\epsilon = 10\%$, the solution is conservative.

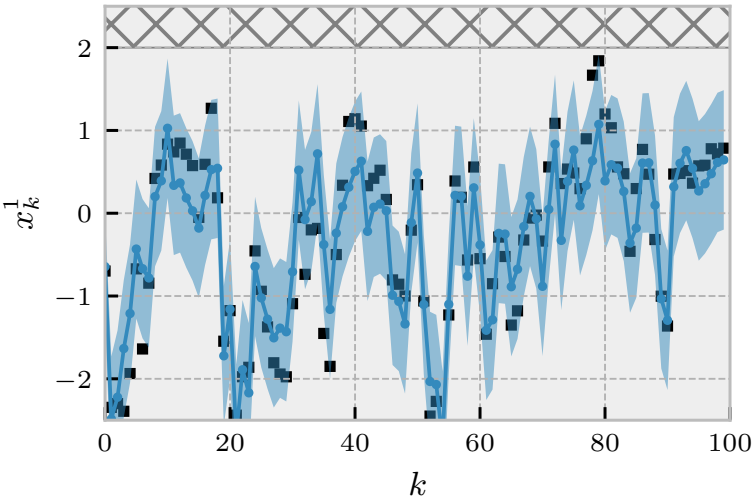

Figure 3.10. The dark blue line is the conditional mean of x_k^1, the shaded area is the two standard deviations about the conditional mean, and the true state is shown as black squares.

3.7 Conclusion

The State Selection Algorithm requires a non-empty set of initial states $\mathbb{X}_0^\epsilon$ to run. The estimator, a particle filter in this case, and the fixed controller κ, have their roles in the feasibility of the algorithm, as we discussed earlier. A particle filter that is prone to depletion restricts the variability of Ξ, and is a poor approximant to the Bayesian filter. While a controller, say $\kappa = 0$, eliminates all control sensitivity to the initial state and to the w'_k, as is shown in the first example of Section 3.6. Such a controller would also extirpate the variation seen in the averaged cost.

While replacing the hard probabilistic constraints with soft ones can avoid infeasibility issues, at least in an algorithmic sense. A more concrete understanding of the problem is required, and hence our future work is towards

- Enriching the proposed distribution of control sequences with more options, potentially by having w'_k of a different statistics than w''_k.
- Identifying more practical characteristics required of the controller κ.
- Exploring parameter estimation for control. This can be done by a simple state augmentation, and the provided framework, whether from the particle filter side, or the State Selection Algorithm, can naturally adapt.

Acknowledgement

Chapter 3, in full, has been submitted for publication of the material as it may appear as: Mohammad S. Ramadan, Robert R. Bitmead, Ke Huang "State estimation for control: an approach for output-feedback stochastic MPC." The dissertation author was the primary investigator and author of this paper.

Appendix A

Derivation of $\mathcal{A}_1$, $\mathcal{A}_2$ and the computation of the controller κ_2 in Chapter 3

Derivation of $\mathcal{A}_1$ and $\mathcal{A}_2$

Substituting the states in terms of their means and errors, from (3.26), in equation (3.23)

$$J_c(x_0') = \mathbb{E}_{x_0''}\mathbb{E}_{W'}\mathbb{E}_{W''}\Bigg(\sum_{k=0}^{N-1} \Big[tr(Q\hat{x}_k''(\hat{x}_k'')^T) + tr(K^T R K \hat{x}_k'(\hat{x}_k')^T) + tr(Q\tilde{x}_k''(\tilde{x}_k'')^T) + \\ + tr(K^T R K \tilde{x}_k'(\tilde{x}_k')^T) \Big] + tr(Q_N \hat{x}_N''(\hat{x}_N'')^T) + tr(Q_N \tilde{x}_N''(\tilde{x}_N'')^T) \Bigg), \tag{A.1}$$

where the cross-terms (means and errors) are ignored due to the states' errors' zero means, eventually by expectation.

The error covariances are not functions of x_0', and thus can be replaced by some constant C in the cost in (A.1) without altering the minimizer. Hence, up to an additive constant

$$J_c(x_0') = \sum_{k=0}^{N-1} \Big[(\hat{x}_k'')^T Q \hat{x}_k'' + (\hat{x}_k')^T K^T R K \hat{x}_k' + (\hat{x}_N'')^T Q_N \hat{x}_N'' \Big] + C, \tag{A.2}$$

where the traces are returned to their quadratic forms. Substituting $\hat{x}_k'$ and $\hat{x}_k''$ from (3.26) in

(A.2) yields

$$J_c(x_0') = \sum_{k=0}^{N-1} \Big[(\hat{x}_0'')^T (F^k)^T Q F^k \hat{x}_0'' + 2(\hat{x}_0'')^T (F^k)^T Q \Psi_{k-1} x_0' + (x_0')^T \Psi_{k-1}^T Q \Psi_{k-1} x_0' + (x_0')^T (F_K^k)^T K^T R K F_K^k x_0' + (\hat{x}_0'')^T (F^N)^T Q F^N \hat{x}_0'' + 2(\hat{x}_0'')^T (F^N)^T Q \Psi_{N-1} x_0' + (x_0')^T \Psi_{N-1}^T Q \Psi_{N-1} x_0' \Big] + C. \tag{A.3}$$

All terms which are constants with respect to x_0' can be added to C to be C_1,

$$J_c(x_0') = \sum_{k=0}^{N-1} \Big[2(\hat{x}_0'')^T (F^k)^T Q \Psi_{k-1} x_0' + (x_0')^T \Psi_{k-1}^T Q \Psi_{k-1} x_0' + (x_0')^T (F_K^k)^T K^T R K F_K^k x_0' + 2(\hat{x}_0'')^T (F^N)^T Q_N \Psi_{N-1} x_0' + (x_0')^T \Psi_{N-1}^T Q_N \Psi_{N-1} x_0' \Big] + C_1, \tag{A.4}$$

or in a compact form, ignoring additive constants

$$J_c(x_0') = (x_0')^T \mathcal{A}_1 x_0' + (\hat{x}_0'')^T \mathcal{A}_2 x_0', \tag{A.5}$$

where

$$\mathcal{A}_2 = \sum_{k=0}^{N-1} \Big[2(F^k)^T Q \Psi_{k-1} \Big] + 2(F^N)^T Q_N \Psi_{N-1}, \tag{A.6}$$

$$\mathcal{A}_1 = \sum_{k=0}^{N-1} \Big[\Psi_{k-1}^T Q \Psi_{k-1} + (F_K^k)^T K^T R K F_K^k \Big] + \Psi_{N-1}^T Q_N \Psi_{N-1}. \tag{A.7}$$

Computation of κ_2

Let V be the value function, the optimal cost, of the infinite-horizon discounted-cost whose stage costs are $\ell_k(x_k, u_k) = x_k^T x_k + u_k^2$ and with a discount factor $\gamma = 0.9$ [8]. We define

κ_2 as the corresponding optimal control. That is, the Dynamic Programming Equation is

$$V(x_k) = x_k^T x_k + \min_{u_k \in \bar{\mathbb{U}}(x_k)} \left\{ u_k^2 + \gamma \mathbb{E}_{w_k} V(f(x_k, u_k, w_k)) \right\}, \tag{A.8}$$

where $\bar{\mathbb{U}}(x_k)$ is the state dependent input constraint set, defined as $\bar{\mathbb{U}}(x_k) = \{u \in \mathbb{U} \mid \mathbb{P}(f(x_k, u_k, w_k) \in \mathcal{L}) < \epsilon\}$.

The controller κ_2 can be computed via the Value Iteration over the finite gridded state space and input space Markov Decision Process, which if the grid size of these spaces is large enough, κ_2 is optimal with respect to the original infinite state-space problem [7].

Following the dynamic programming approach in [44], uniformly randomized grids of 4000 points over $[-10, 10] \times [-5, 15]$ in the state space and 50 points in $[-3, 3]$, the control space $\mathbb{U}$, are generated. The continuous probability in the definition of $\bar{\mathbb{U}}(x)$ is replaced by discrete over the points of the grid which are inside $\mathcal{L}$, and the expectation in (A.8) by its sample average over the grid. Value Iteration was conducted over the grid until convergence, which is guaranteed. The resulting optimal control

$$\kappa_2(x_k) = \arg\min_{u_k \in \bar{\mathbb{U}}(x_k)} \left\{ u_k^2 + \gamma \mathbb{E}_{w_k} V(f(x_k, u_k, w_k)) \right\}, \tag{A.9}$$

can be computed by replacing the expectation with its sample average over the grid. A point ζ on the grid such that $\kappa(\zeta)$ is infeasible is included into the grid defining $\mathcal{L}$.

After evaluating κ_2, it is then fitted into a piecewise linear surface over the region $[-10, 10] \times [-15, 5]$ of the state space, and $\kappa_2(x)$ is then acquired by 2D linear interpolation.

Bibliography

[1] Brian D.O. Anderson and John B. Moore. *Optimal Control: Linear Quadratic Methods*. Prentice-Hall, 1989.

[2] Karl J Åström. Automatic control—the hidden technology. In *Advances in Control: highlights of ECC'99*, pages 1–28. Springer, 1999.

[3] Karl J Åström. *Introduction to stochastic control theory*. Courier Corporation, 2012.

[4] Karl Johan Åström and Anders Helmersson. Dual control of an integrator with unknown gain. *Computers & Mathematics with Applications*, 12(6):653–662, 1986.

[5] Yaakov Bar-Shalom. Stochastic dynamic programming: Caution and probing. *IEEE Transactions on Automatic Control*, 26(5):1184–1195, 1981.

[6] Alberto Bemporad, Manfred Morari, Vivek Dua, and Efstratios Pitsikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

[7] Dimitri Bertsekas. Convergence of discretization procedures in dynamic programming. *IEEE Transactions on Automatic Control*, 20(3):415–419, 1975.

[8] Dimitri Bertsekas. *Dynamic programming and optimal control: Volume I*, volume 1. Athena scientific, 2012.

[9] Mark Cannon, Qifeng Cheng, Basil Kouvaritakis, and Saša V Raković. Stochastic tube mpc with state estimation. *Automatica*, 48(3):536–541, 2012.

[10] Mark Cannon, Basil Kouvaritakis, Saša V Raković, and Qifeng Cheng. Stochastic tubes in model predictive control with probabilistic constraints. *IEEE Transactions on Automatic Control*, 56(1):194–200, 2010.

[11] David N. DeJong, Roman Liesenfeld, Guilherme V. Moura, Jean-François Richard, and Hariharan Dharmarajan. Efficient likelihood evaluation of state-space representations. *Review of Economic Studies*, 80(2):538–567, 2013.

[12] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[13] Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000.

[14] Marcello Farina, Luca Giulioni, Lalo Magni, and Riccardo Scattolini. A probabilistic approach to model predictive control. In *52nd IEEE Conference on Decision and Control*, pages 7734–7739. IEEE, 2013.

[15] Aleksandr Aronovich Fel'dbaum. *Optimal Control Systems*. Elsevier, 1966.

[16] Tor Aksel N Heirung, Joel A Paulson, Jared O'Leary, and Ali Mesbah. Stochastic model predictive control—how does it work? *Computers & Chemical Engineering*, 114:158–170, 2018.

[17] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. In *The collected works of Wassily Hoeffding*, pages 409–426. Springer, 1994.

[18] Nikolas Kantas, Arnaud Doucet, Sumeetpal S. Singh, Jan Maciejowski, and Nicolas Chopin. On particle methods for parameter estimation in state-space models. *Statistical Science*, 30(3):328–351, 2015.

[19] Nikolas Kantas, JM Maciejowski, and A Lecchini-Visintini. Sequential monte carlo for model predictive control. In *Nonlinear model predictive control*, pages 263–273. Springer, 2009.

[20] S Sathiya Keerthi and Elmer Grant Gilbert. Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations. 1988.

[21] Panqanamala Ramana Kumar and Pravin Varaiya. *Stochastic systems: Estimation, identification, and adaptive control*. SIAM, 2015.

[22] HC La, A Potschka, JP Schlöder, and HG Bock. Dual control and information gain in controlling uncertain processes. *IFAC-PapersOnLine*, 49(7):139–144, 2016.

[23] Kenneth Lange. A gradient algorithm locally equivalent to the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(2):425–437, 1995.

[24] Wilbur Langson, Ioannis Chryssochoos, SV Raković, and David Q Mayne. Robust model predictive control using tubes. *Automatica*, 40(1):125–133, 2004.

[25] Sylvain Le Corff and Gersende Fort. Convergence of a particle-based approximation of the block online expectation maximization algorithm. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 23(1):1–22, 2013.

[26] Baibing Li, Cunjia Liu, and Wen-Hua Chen. An auxiliary particle filtering algorithm with inequality constraints. *IEEE Transactions on Automatic Control*, 62(9):4639–4646, 2016.

[27] Mengheng Li and Marcel Scharth. Leverage, Asymmetry, and Heavy Tails in the High-Dimensional Factor Stochastic Volatility Model. *Journal of Business & Economic Statistics*, pages 1–17, 2020.

[28] Andreas Lindholm and Fredrik Lindsten. Learning dynamical systems with particle stochastic approximation EM. *arXiv preprint arXiv:1806.09548*, 2018.

[29] Matthias Lorenzen, Fabrizio Dabbene, Roberto Tempo, and Frank Allgöwer. Constraint-tightening and stability in stochastic model predictive control. *IEEE Transactions on Automatic Control*, 62(7):3165–3177, 2016.

[30] James Luedtke and Shabbir Ahmed. A sample approximation approach for optimization with probabilistic constraints. *SIAM Journal on Optimization*, 19(2):674–699, 2008.

[31] Sheheryar Malik and Michael K. Pitt. Particle filters for continuous likelihood evaluation and maximisation. *Journal of Econometrics*, 165(2):190–209, 2011.

[32] David Q Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967–2986, 2014.

[33] David Q Mayne, María M Seron, and SV Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2):219–224, 2005.

[34] Xiao-Li Meng and Donald B. Rubin. Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, 80(2):267–278, 1993.

[35] Ali Mesbah. Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems Magazine*, 36(6):30–44, 2016.

[36] Lyudmila Mihaylova, Avishy Y. Carmi, François Septier, Amadou Gning, Sze Kim Pang, and Simon Godsill. Overview of Bayesian sequential Monte Carlo methods for group and extended object tracking. *Digital Signal Processing*, 25:1–16, 2014.

[37] Todd K. Moon. The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, 13(6):47–60, 1996.

[38] Roger Penrose. *The road to reality*. Random house, 2006.

[39] Michael K. Pitt and Neil Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599, 1999.

[40] George Poyiadjis, Arnaud Doucet, and Sumeetpal S. Singh. Particle approximations of the score and observed information matrix in state space models with application to parameter estimation. *Biometrika*, 98(1):65–80, 2011.

[41] Herbert E. Rauch, F. Tung, and Charlotte T. Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8):1445–1450, 1965.

[42] Benjamin Recht. A tour of reinforcement learning: The view from continuous control. *Annual Review of Control, Robotics, and Autonomous Systems*, 2:253–279, 2019.

[43] Sidney Resnick. *A probability path*. Springer, 2019.

[44] John Rust. Using randomization to break the curse of dimensionality. *Econometrica: Journal of the Econometric Society*, pages 487–516, 1997.

[45] Henning Schlüter and Frank Allgöwer. Stochastic model predictive control using initial state optimization. *arXiv preprint arXiv:2203.01844*, 2022.

[46] Thomas B. Schön, Adrian Wills, and Brett Ninness. System identification of nonlinear state-space models. *Automatica*, 47(1):39–49, 2011.

[47] Martin A Sehr and Robert R Bitmead. Particle model predictive control: Tractable stochastic nonlinear output-feedback mpc. *IFAC-PapersOnLine*, 50(1):15361–15366, 2017.

[48] François Septier, Sze Kim Pang, Simon Godsill, and Avishy Carmi. Tracking of coordinated groups using marginalised MCMC-based particle algorithm. In *2009 IEEE Aerospace Conference*, pages 1–11. IEEE, 2009.

[49] Jennie Si, Andrew G Barto, Warren B Powell, and Don Wunsch. *Handbook of learning and approximate dynamic programming*, volume 2. John Wiley & Sons, 2004.

[50] Charlotte Striebel. Sufficient statistics in the optimum control of stochastic systems. *Journal of Mathematical Analysis and Applications*, 12(3):576–592, 1965.

[51] Vladislav Z.B. Tadić and Arnaud Doucet. Asymptotic properties of recursive particle maximum likelihood estimation. *IEEE Transactions on Information Theory*, 67(3):1825–1848, 2020.

[52] Emanuel Todorov et al. Optimal control theory. *Bayesian brain: probabilistic approaches to neural coding*, pages 268–298, 2006.

[53] Edison Tse, Yaakov Bar-Shalom, and Lewis Meier. Wide-sense adaptive dual control for nonlinear stochastic systems. *IEEE Transactions on Automatic Control*, 18(2):98–108, 1973.

[54] Emilian R. Vankov, Michele Guindani, and Katherine B. Ensor. Filtering and estimation for a class of stochastic volatility models with intractable likelihoods. *Bayesian Analysis*, 14(1):29–52, 2019.

[55] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8:279–292, 1992.

[56] Greg C.G. Wei and Martin A. Tanner. A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85(411):699–704, 1990.

[57] C.F. Jeff Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1):95–103, 1983.

[58] Jun Yan and Robert R Bitmead. Incorporating state estimation into model predictive control and its application to network traffic control. *Automatica*, 41(4):595–604, 2005.