# Optimal Population Coding of Dynamic Stimuli

vorgelegt von
Alex Kunze Susemihl
geboren in São Paulo, Brasilien

von der Fakultät IV - Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften
- Dr.-rer.-nat. -

genehmigte Dissertation

Promotionsausschuss:

| | |
|---|---|
| Vorsitzender: | Prof. Dr. Klaus Obermayer |
| Gutachter: | Dr. Jakob Macke |
| Gutachter: | Prof. Dr. Manfred Opper |
| Gutachter: | Prof. Dr. Ron Meir |

Tag der wissenschaftlichen Aussprache: 23. Januar 2015

Berlin 2015

Alex Kunze Susemihl

Technische Universität Berlin
Bernstein Center for Computational Neuroscience

# Optimal Population Coding of Dynamic Stimuli

# *Abstract*

Organisms are routinely faced with the task of making decisions based on partial, unreliable information. Furthermore, not only is the environment noisy, but the computational units organisms employ themselves are often faulty and noisy, such as the neurons of the animal nervous system. There is a crucial difference, though: while the organism can only act indirectly on its environment, its sensory organs are under its direct control through adaptation. Here I thus consider how a population of neurons can organise itself to allow for optimal information processing at its output. I will focus on a model of a rapidly changing environment, which forces the organism to make short-time decisions.

In that setting I have provided a number of novel results, extending the framework of population coding to a fully dynamic setting, where a changing environment is decoded from spike trains in an online fashion. To do so, I have drawn from the theory of stochastic filtering of doubly stochastic point processes. For a dense population of neurons, this reduces to a Gaussian process regression problem, for which I have obtained closed form relations for the error distribution. The formalism developed also lends itself to a direct extension to a control-theoretical setting. Thus I have also developed a criterion for optimal coding in the case of control problems, and have compared this to the case of estimation previously developed. Interestingly, I have been able to show that the different objective functions for control and estimation lead to different optimal encoders.

The study of optimal coding in populations of neurons is an active fertile area of research. Here I have extended a number of previous findings to the case of online decoding of dynamic stimuli from point processes. Parallel to that, I have discussed the issue of optimal coding for control, and how it relates to the study of optimal coding in the estimation case.

# *Zusammenfassung*

Lebende Organismen müssen ständig Entscheidungen treffen, die sich auf partieller, ungewisser Information basieren. Nicht nur des Organismus Umfeld ist unsicher, sondern auch die komputationellen Einheiten, die es zur Bildung einer Entscheidung anwendet, wie zum Beispiel, Neuronen in Nervensystemen. Immerhin besteht zwischen den beiden ein wichtiger Unterschied: auf die Umwelt kann das Organismus nur indirekt agieren, während die Neuronen direkt adaptiv eingestellt werden können. Ich befasse mich hier mit der Frage, wie Populationen von Neuronen sich organisieren können um die Informationsverarbeitung seiner Ausgänge zu erleichtern. Darin fokussiere Ich mich auf dynamischen Stimuli, die schnelle Entscheidungen erbitten.

In diesem Umfeld habe Ich mehrerer neue Ergebnisse erreicht, und das Feld der Populationskodierung zu einem dynamischen Kontext ausgearbeitet, in dem eine dynamische Umwelt *online* dekodiert wird, also in Echtzeit. Dazu habe Ich die Theorie der Filterung doppelstochastischer Punktprozesse angewendet. Für dichte Populationen von Neuronen kann das als ein Gaussprozessregressionsproblem formuliert werden, für welchen Ich ein geschlossenen Ausdruck für die Verteilung der Fehler fand. Das erarbeitete Formalismus erlaubt eine direkte Erweiterung zu einem kontrolltheoretischen Kontext, und Ich habe die Ergebnisse in der kontrolltheoretischen Formulierung mit den Ergebnissen in der stochastischen Filterung verglichen. Interessanterweise, war es mir möglich zu zeigen, dass beide Aufgaben zu unterschiedlichen optimalen Kodierungsstrategien führen.

Das Feld der optimalen Kodierung in neuronalen Populationen ist ein aktives Forschungsfeld. Ich habe hier mehrere Ergebnisse zum Fall von dynamischen Stimuli erweitert. Parallel habe Ich diese Ergebnisse mit Kontrol-theoretischen Ergebnissen verglichen und diskutiert welche Implikationen das hat.

# *Acknowledgements*

In the media and among friends, it seems a particular narrative of personal success is becoming more and more popular. I mean the narrative of *having made it on my own*, or *building my own success* and *being responsible for my own success*. I personally find this kind of notion arrogant if not ludicrous, and operate under no such assumption. In the timeless words of Billy Shears, *I get by with a little help from my friends*. Let me then name a few of them.

None of this, my coming to Berlin, my doctoral studies, would have happened if I had not met Manfred Opper, and I have come to consider him a friend as well as a mentor. Our discussions have always touched on fascinating subjects, such as the tale of an austrian nobleman who came to be executed as the emperor of Mexico, the story of the siege of Jerusalem and classic rock facts, occasionally mentioning functional analysis, function-valued stochastic processes and information theory. I also wish to thank the KI group he leads, Florian, Cordula, Andreas, Philipp and more recently Ludovica, for providing a supportive, relaxed and fun work environment with a lot of stimulating discussions.

There are many friends I have made in Berlin, and though it is hard to single one out, Chris is surely the one who shared most of my experience here. In the meantime we have shared a research project, become friends and become parents, and it has been a great joy to share these experiences with him, with his awesome wife Katrin and his even more awesome daughter Florence. Thank you very much guys, you are missed.

One of the greatest things about my time in Berlin was the opportunity to meet so many different people, and learning how much we share. Sinem has taught me how similar brazilian and turkish people are, and has been a great research partner and friend. Our lunchtimes will be missed, Sinem. Though it was hardly a surprise, Tommaso also taught me how much Italy has in common with Brazil, and the warmth of your friendship has made me feel at home away from home. There were so many others it is hard to name them all, but I can not refrain from mentioning Joachim, always fun company with cool board games and weird factoids, Fred, the BCCN's smiling DJ, always uplifting and bringing in some se-

riously cool disco songs, and Philippe, our awesome canadian who can easily go five hours on a single conversation and will not let the night end a second too early.

Someone once told me: *Friends come and go, enemies accumulate.* I must be a very lucky person then, since, unless I am completely oblivious to it, I have managed to avoid making blood feuds while holding on to some great friends from older times. Marcos and Flavia, whom we followed to Berlin, have always been a source of support, and we have not met nearly as much as I'd like to in these last years. Domingos, who followed us to Berlin to live the student life, remains a great friend, and it was a great pleasure being able to share some of my Berlin time with you. And finally, Igor, who from afar has still kept us close to his daily life. Thank you all.

Naturally, a lot of my time in Berlin has actually been spent with academic work, and I would like to thank my collaborators. Ronny has always been a great collaborator and advisor, and I sincerely do not believe my papers would have made as much sense if he hadn't helped set the context and organise our ideas into a coherent body of research. I thank him for the collaboration and for the patience. Martin, Chris' supervisor, has also been a great advisor, and has helped us put a research project on firm standing and framing it in a neuroscientific context. I am, however, most in debt to Vanessa Casagrande for making my work possible. Besides helping in navigating the german bureaucracy, securing soft skill offerings, organising retreats and research visits, you have always been a source of support and guidance for me and surely for most of the students in the GRK. Thank you for that and for being a great friend.

Most of all, though, I am in debt for my family for supporting me and making this possible. First of all, I would like to thank my wife, Fernanda, for joining me in moving across the globe and supporting me unconditionally. That is more than I could have hoped for, but you went overboard and presented me with the greatest gift I could imagine, our little Hugo, and for that I am forever grateful to you. You two are my inspiration. My parents and my brother have always been there for me, and I am forever thankful for your support, your love and the interest you take in whatever crazy project I decide to drag myself into. If I have ever felt safe to try my luck in the world, it was because I knew I had a safe haven to turn back to.

# *Contents*

# 1
# *Introduction*

> A wing would be a most mystifying structure if one did not know that birds flew.
>
> Horace Barlow

NEUROSCIENCE AS A WHOLE is concerned with the function of the nervous system. More precisely, it asks a very simple question: *What is the brain doing?*[1] The simplicity with which humans and animals perform in their environment makes it almost unnatural to ask how their brains enable these behaviours. Many actions are performed so naturally, that it is often hard to explain to laymen the complexity involved in preparing even the simplest actions, such as saccades or walking. Although one can not realistically expect to answer that question in a general fashion, I will try to touch upon a number of points which shed light on some aspects of the nervous system and provide us with a *guiding principle* to understand what the brain is doing, why and possibly how.

[1] Or alternatively: *What is the nervous system doing?*

Neuroscience was born as a branch of biology, and although it is now often thought of as an interdisciplinary science in itself, its objects of study are still to the largest extent biological systems. Theodosius Dobzhansky published an influential essay in 1973, entitled *Nothing in biology makes sense except in the light of evolution*,[2] which defends exactly that point. Though the theory of evolution through natural selection has been reviewed and revisited constantly since its proposal, it remains the central pillar of biological sciences. As such, neuroscience must also view its objects of study through the lense of evolution. More specifically, we can then ask ourselves *What evolutionary advantage would this brain bring to an individual?* instead of *Why is the brain this way?* That being said, there are caveats in the case of neuroscience. For one, the brain is capable of plasticity and adaptation unthinkable for other organs, and so one can not expect to understand the functionality of the brain as a function of its environment as simply as the shape of bird beaks can be understood as

[2] Dobzhansky, T. (1973). Nothing in Biology Makes Sense Except in the Light of Evolution. *The American Biology Teacher*, 35(3):125–129

an adaptation to their preferred diet. Furthermore, the brain controls all of the motor and perceptual apparatus, having a multitude of uses and purposes, unlike simpler organs.

A growing body of literature supports the idea that the brain's organisation is adapted to the environment it operates in. More precisely, the response properties of many sensory areas in the animal brain are such that they encode their natural stimuli in an optimal fashion. Visual receptive fields in V1 resemble the independent components of visual images,[3] the principal components of natural sounds resemble the auditory responses in auditory fibers,[4] the colour sensitivity of retinal cells can be traced back to the colour spectrum in primate's and fish's environment,[5] and so on. My main goal in this thesis is to extend this kind of approach to a dynamic setting, in which the whole spike train is used to encode the environment, instead of graded responses or spike counts. In that sense, I will be looking at a simple model of a sensory neural population which responds to a dynamic stimulus like a Poisson process with a time-dependent rate. By considering the task of reconstructing the encoded stimulus from the spike train of the population, I will study the mean squared error as a measure of the performance of the encoding population. Though this thesis is mostly theoretical, I hope it will lay the foundation to the study of optimal population coding using full spike trains in a true time-depedendent fashion. In the remainder of this chapter I will better contextualise my goals and the tools I will employ.

[3] Bell, A. J. and Sejnowski, T. J. (1997). The independent components of natural scenes are edge filters. *Vision research*, 37(23):3327–3338; and Olshausen, B. A. and Field, D. J. (1996). Natural image statistics and efficient coding. *Network*, 7(2):333–339

[4] Lewicki, M. S. (2002). Efficient coding of natural sounds. *Nature neuroscience*, 5(4):356–63

[5] Atick, J. J. (1992). Could information theory provide an ecological theory of sensory processing? *Network: Computation in neural systems*, 3(2):213–251

## *1.1 Hierarchical Organisation of the Brain and the Feedforward Paradigm*

One of the most distinguishing properties of the mammalian brain, and of most neural systems in nature, is its hierarchical organisation. The human cortex has a very marked organisation with clear functional units and very distinct connectivity patterns within functional units and between them.[6] A rough view of the flow of information in the sensory areas of the human brain can be seen in figure 1.1. According to this paradigm, information about the environment enters the nervous system through the primary sensory areas, which code for simple aspects of the environment such as edges of visual shapes or particular sound frequencies. These areas then transmit that information to higher brain areas, often called secondary and tertiary areas. The secondary and tertiary areas then process the input further, integrating information within and between sensory modalities. The motor cortex proceeds in a similar fashion but in the opposite direction. The tertiary motor areas receive information from

[6] Kandel, E. R., Schwartz, J. H., Jessell, T. M., et al. (2000). *Principles of neural science*, volume 4. McGraw-Hill New York; and Bear, M. F., Connors, B. W., and Paradiso, M. A. (2007). *Neuroscience*, volume 2. Lippincott Williams & Wilkins

higher sensory areas and code for high-level aspects of motor control, such as complex movements, goals and integrated plans. Downstream are the secondary and primary areas, which code for simpler aspects of motor control, with activity in the primary motor cortex often having a simple relation to the movement of joints or limbs.[7]

More interestingly, these areas are anatomically organised in a very distinctive way. The sensory areas are found in the posterior part of the brain, with the primary areas being found further towards the back and the tertiary areas being found near the central sulcus of the brain. The motor areas, in contrast, are found in the frontal part of the brain, the tertiary areas towards the front and the primary motor cortex being located on the frontal side of the central sulcus of the brain.

[7] The phrasing *downstream* and *upstream* are used frequently in neuroscience and are derived from this view of information flow in the brain. The furthest *upstream* areas would be the sensory organs, and the furthest *downstream* areas would be the motor organs. In that sense, the secondary visual cortex V2 is downstream from the primary visual cortex V1, but upstream from the prefrontal cortex or the primary motor cortex.

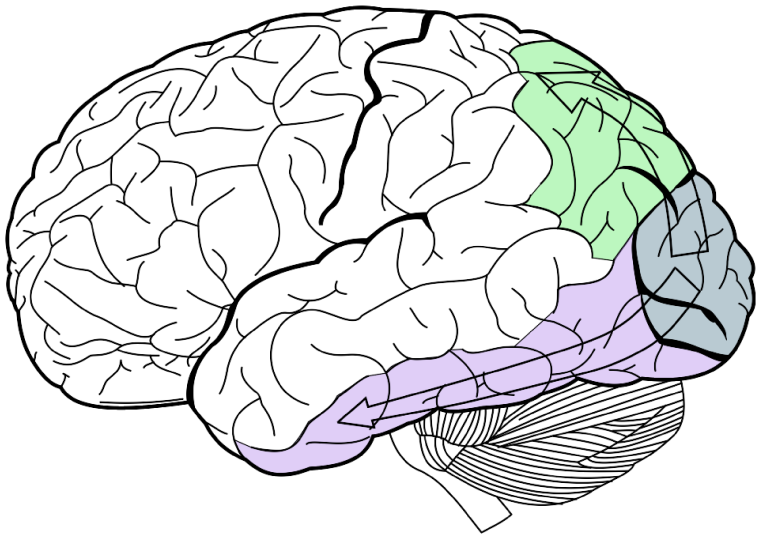

Figure 1.1: The feedforward view of the visual sensory pathway: The two main visual pathways, the ventral (purple) and dorsal (green) pathways are shown. The feedforward paradigm views the information flowing from the primary areas towards the right to the higher areas toward the middle. Figure from `http://commons.wikimedia.org/wiki/File:Ventral-dorsal_streams.svg`.

The finding by Thorsten Wiesel and David H. Hubel that neurons in the primary visual cortex fire specifically in response to certain visual patterns presented in certain areas of the visual field was instrumental to this understanding of the brain, as it was the first to clearly identify neurons that code selectively for simple features of visual stimuli.[8] This view of information processing in the mammalian brain can be summarised in the so-called *feedforward paradigm*: The brain is divided in functional units, which receive input from upstream units, integrate and process that input and then relay the results to downstream units. This has been a very influential idea in systems neuroscience, and though it has come under a lot of criticism recently,[9] a lot of work still bases its assumptions on this paradigm. A fresher view of the functional connections in the visual pathways of the human brain can be seen in figure 1.2, from which it is immediately clear that the simple feedforward view of the brain is somewhat

[8] Hubel, D. H. and Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148(3):574

[9] Gilbert, C. D. and Li, W. (2013). Top-down influences on visual processing. *Nature Reviews Neuroscience*, 14(5):350–363

outdated. Even the earliest sensory areas have been shown to be modulated by a number of higher order factors, such as attention and task-related biases.

The impact of the feedforward paradigm can still be seen in other fields as well. In Machine Learning, for example, feedforward neural networks modelled after the feedforward view of the brain's organisation are still a very active area of research and rank among the most powerful algorithms in the field.[10]

[10] Bengio, Y. (2009). Learning deep architectures for A.I. *Foundations and trends in Machine Learning*, 2(1):1–127

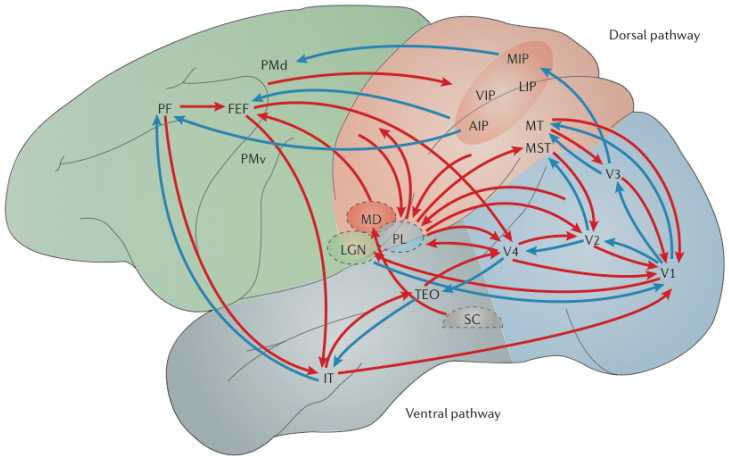


Figure 1.2: Feedforward and Feedback pathways carrying top-down information in the macaque brain: In blue are shown feedforward connections between areas of the visual pathways. In red are shown feedback connections conveying top-down information to upstream sensory areas. Figure from (Gilbert and Li, 2013)

I have briefly illustrated the feedforward paradigm, which gives us a handle on the brain's organisation. It is by no means a general explanation of the brain's workings, but it gives one a structured framework to think about its form and function. It does not, however, specify the workings of every cortical area. Let us consider for example the primary visual cortex (V1). It receives inputs from the retina through the Lateral Geniculate Nucleus and its neurons respond to visual stimuli according to spatiotemporal filters of the presented stimulus. From a purely conceptual point of view, V1 transform the representation of the visual stimulus from a simple ON- and OFF-cell representation found in the retina to a more complex representation in terms of spatiotemporal Gabor filters and similar response functions. The responses of retinal photoreceptors are thus pooled and processed together to give rise to more complex features. One can then ask how these computations are performed and what the ideal way of performing this would be.

Neurons are inherently noisy cells, and their responses to the same stimulus presentation are often very variable.[11] One can then ask how the subsequent stages of information processing in the brain deal with the noise present in its input. From a theoretical viewpoint, one can further ask how a particular cortical area should be organised to facilitate the information processing by downstream areas.[12] This is hard to

[11] Faisal, A. A., Selen, L. P., and Wolpert, D. M. (2008). Noise in the nervous system. *Nature Reviews Neuroscience*, 9(4):292–303

[12] The response of a population of neurons to a particular aspect of the environment is often called a *population code*, and I will use this term throughout the text.

do without a clear view of the exact computational process being performed by a neural system, but one can resort to a number of theoretical frameworks to bypass this problem. If one can agree on a general computational task a population of neurons is performing (i.e.: estimating the direction of motion of the visual field or detecting the presence of an auditory stimulus masked in noise), we can use statistical tools to provide bounds on the performance of a given population code. For example, assuming neurons in V1 seek to estimate the direction of movement in a patch of the visual field, one can ask: What is the best arrangement of the receptive fields for the neurons projecting into V1 if that population is to detect the direction of moving gratings? I have repeatedly used the phrase *process information* but the meaning of this is far from obvious. Let me specify what is usually meant by information in the field of computational neuroscience.

## *1.2 Quantifying Information*

Information is a widely used term, but one usually has a very vague conception of what it means. Having information about an event usually means having the means to describe, reconstruct or distinguish that particular event. There are many ways of defining information, and I will shortly consider three different approaches.

### *Information Theory and Mutual Information*

The most common definition of information is probably the one from Shannon's information theory.[13] Information theory defines the information associated with a random event as the logarithm of its inverse probability. So, for a random variable $X$ taking values $x \in \mathcal{A}_X$ with probability $P_X(x)$, the information of a particular outcome $x_i$ would be

$$I(x_i) = \log\left(\frac{1}{P_X(x_i)}\right) = -\log P_X(x_i).$$

This is often called the surprise associated with an event as well, as very probable outcomes are not very informative. The limit of zero information is an event one is absolutely certain about, therefore observing it conveys no information at all. The opposing case are very rare events, which carry a lot of information.[14]This might seem a rather unusual concept of information at first glance, but it is a very useful one, having a very wide applicability.[15] One can then also define the entropy of a distribution over $X$ as the average information

[13] Shannon, C. E. (1948). The mathematical theory of communication. 1963. *Bell System Technical Journal*, 27:379–423 and 623–656

[14] One is almost absolutely certain that days will turn into nights and vice-versa. If at a given morning, I observe the dawn, this does not hold a lot of information. If at a given morning the dawn fails to arrive, that would be an informative event, possibly a harbinger of a storm or eclipse.

[15] Most forms of modern communication systems depend in some way on developments of information theory. Error-correcting codes have made digital communication and information storage possible by robustly dealing with the inherent noise in physical systems. Cryptography and compression methods also draw from the conclusions of information theory, among many others.

gained from observing the outcome of $X$. This gives

$$H[X] = -\sum_{x \in \mathscr{A}_X} P_X(x) \log P_X(x),$$

which is a measure of the disorder or uncertainty of the random variable $X$. This gives one a very interesting connection to the field of statistical mechanics, where entropy plays a central role. It does not fit perfectly into the colloquial meaning of information, however, as the concept of information is mostly referential, in the sense that information is about something.

If $X$ is a continuous random variable, one needs to formulate these ideas a bit more precisely. Say that $X$ takes values $x \in \mathscr{A}_X$, where $\mathscr{A}_X$ in turn is a subset of $\mathbf{R}^n$. The probability must now be defined in terms of a probability density. One has then, for a continuous random variable $X$, the probability of $X$ taking a value in a set $\mathscr{B} \subset \mathscr{A}_X$[16]

$$P(X \in \mathscr{B}) = \int_{\mathscr{B}} dx P_X(x).$$

In this thesis, I will mostly consider probability densities as the objects of interest, but I trust the reader to notice when I am working with discrete distributions from the context. For the continuous case we can then define

$$H[X] = -\int_{\mathscr{A}_X} dx P_X(x) \log P_X(x).$$

This is called the continuous or differential entropy and it can be positive or negative. This makes it difficult to interpret its value directly as a measure of disorder or uncertainty. Technically speaking, the differential entropy is not the continuous limit of the discrete entropy, as the continuous limit would lead to the Riemann sum

$$-\sum_{x \in \mathscr{A}_x} P_X(x) dx \log\left(P_X(x) dx\right),$$

which clearly diverges in the limit $dx \to 0$. The Kullback-Leibler divergence, or relative entropy, between two distinct distributions of $X$, however, has a well defined limit in the continuous case. Given two distributions $P(x)$ and $Q(x)$ for a discrete random variable $X$, we have the KL-divergence

$$KL[P\|Q] = \sum_{x \in \mathscr{A}_X} P(x) \log\left[\frac{P(x)}{Q(x)}\right].$$

It is easy to see that taking the continuous limit we obtain for densities the KL-divergence

$$KL[P\|Q] = \int_{x \in \mathscr{A}_X} dx P(x) \log\left[\frac{P(x)}{Q(x)}\right].$$

[16] I am here using the Riemann integral to formulate the probability, placing some restrictions on the nature of the set $\mathscr{B}$. For safety, one can assume that $\mathscr{B}$ is compact. These can be further loosened by writing the probability in terms of the Lebesgue measure as

$$P(X \in \mathscr{B}) = \int_{\mathscr{B}} d\mu(x) P_X(x),$$

but this is not necessary to the treatment developed herein.

The mutual information between two random variables $X$ and $Y$ quantifies the dependence between them. The Mutual information between $X$ and $Y$ is defined as

$$I(X,Y) = \sum_{x \in \mathscr{A}_X} \sum_{y \in \mathscr{A}_Y} P_{XY}(x,y) \log\left(\frac{P_{XY}(x,y)}{P_X(x)P_Y(y)}\right),$$

where $P_{XY}$ is the joint distribution of $X$ and $Y$. Note that the mutual information is the KL-divergence between the joint distribution $P_{XY}(x,y)$ and the product of the marginals $P_X(x)P_Y(y)$, and is therefore always positive. The mutual information between two random variables will be zero if and only if the two random variables are statistically independent. In that sense, the mutual information quantifies how dependent the two variables are and therefore how much one can know about one from observing the other. This is further clarified by rewriting the mutual information as

$$I(X,Y) = \sum_{y \in \mathscr{A}_Y} P_Y(y) \sum_{x \in \mathscr{A}_X} P_X\left(x|Y=y\right) \log\left(\frac{P_X\left(x|Y=y\right)}{P_X(x)}\right) = \boldsymbol{E}_Y\left(H\left[X\right] - H\left[X|Y=y\right]\right),$$

where $P_X\left(x|Y=y\right)$ is the conditional distribution of $X$ conditioned on the outcome of $Y$ being equal to $y$ and $H[X|Y=y]$ is the entropy of that distribution. So the mutual information is equal to the average reduction in the entropy of $X$ caused by an observation of $Y$. As mentioned before, the entropy is a measure of the uncertainty or disorder of a random variable, so the mutual information quantifies how much the uncertainty in $X$ is reduced by observing $Y$. This is already much nearer to our colloquial understanding of information. Take for example, the event $X$ to be a measurement of the atmospheric pressure, and $Y$ to be the occurrence of a rainstorm. Clearly, the uncertainty about the occurrence of a rainstorm is decreased by a measurement of the atmospheric pressure, and the mutual information will have a non-zero value. If $X$ were the outcome of a fair coin toss, there would be reason to believe that the mutual information between and $X$ and $Y$ should be zero.

In a neuroscientific context, one can think of one of the random variables ($X$) as representing the environment or the input to a neural population and the other variable as representing the population's response ($Y$). The mutual information then quantifies how much the population's response reduces the uncertainty about the system's state. The distribution $P_X(x)$ represents the natural distribution of stimuli in the environment and $P_Y(y|X=x)$ gives the distribution of population responses $Y$ given the environment's state $X=x$. The mutual information is symmetric in its arguments and can also be thought of as the reduction in uncertainty in the neuron's responses upon the observation of the environment's

state

$$I(X,Y) = \sum_{x \in \mathscr{A}_X} P_X(x) \sum_{y \in \mathscr{A}_Y} P_Y(y|X=x) \log\left(\frac{P_Y(y|X=x)}{P_Y(y)}\right) = \boldsymbol{E}_X\left(H[Y] - H[Y|X=x]\right).$$

The mutual information and entropy of neural responses are a widely employed measure of a neural code's quality and has been often used to establish the optimality of experimentally measured coding strategies,[17] as well as to explain general features of neural systems.[18]

I have only briefly introduced the concepts from information theory, but it should be noted that a number of theoretical results reinforce the interpretation of the mutual information as a measure of the information content of a code. Shannon's theorems are often defined in terms of the capacity of a channel, which is specified by a distribution of messages $Y$ conditioned on the source $X$. The capacity of a channel given by a distribution $P_Y(y|X=x)$ is then

$$C = \max_{P_X(x)} I(X,Y).$$

Shannon's noisy channel coding theorem, for example, states that it is possible to transmit messages through the given channel at a rate $R < C$ with a vanishingly small error in the limit of long messages.[19]

In the context of optimal neural coding, one could then experimentally measure the distribution of certain features in natural stimuli and seek out the neural response distribution $P_Y(y|X=x)$ which maximises the mutual information between the environment and the response. I will consider a simple example from the literature.

The large monopolar cells (LMC's) of the visual system of the blowfly respond to light contrast on a particular area of the visual field with a graded potential response. In (Laughlin, 1981), the author explored what the best way of organising these responses would be according to information theory.[20] Since the mutual information gives the information content of a response $Y$ about a stimulus $X$, it makes sense to maximise the mutual information between them. Given an environmental distribution of contrasts $P_X(x)$ one must choose a distribution $P_Y(y|X=x)$ that maximises the mutual information between the stimulus and the response. Furthermore, assuming the response $Y$ is a deterministic function of the stimulus $g(X)$, the conditional entropy $H[Y|X=x]$ will be zero and one is left with maximising the entropy of $Y$.[21] The distribution which maximises the entropy over a finite domain without any further constraints is the uniform distribution (see appendix A.1) and one is led to conclude that $P_Y(y)dy = \beta dy$ gives the optimal response distribution,

[17] Laughlin, S. (1981). A simple coding procedure enhances a neuron's information capacity. *Z. Naturforsch*, 36(c):910–912

[18] Tkacik, G., Prentice, J. S., Balasubramanian, V., and Schneidman, E. (2010). Optimal population coding by noisy spiking neurons. *Proceedings of the National Academy of Sciences of the United States of America*, 107(32):14419–24

[19] MacKay, D. J. C. (2003). *Information theory, inference and learning algorithms*. Cambridge university press; and Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*, volume 6 of *Wiley Series in Telecommunications*. Wiley

[20] This analysis appeared first in (Laughlin, 1981), but the treatment shown here is taken from (Atick, 1992).

[21] This is somewhat more delicate in the continuous case, as the entropy of a Dirac delta distribution is undefined. Nevertheless, the conditional entropy $H[Y|X=x]$ is independent of the specific choice of $g(X)$ as long as $g$ is one-to-one, so I will ignore this issue. The whole analysis can also be done for discrete response levels, which bypasses this problem completely.

where $\beta$ is a normalisation constant. But $y = g(x)$, so

$$P_Y(y)dy = P_Y(y)\frac{dg}{dx}dx = P_X(x)dx,$$

and finally

$$g(y) = \frac{1}{\beta}\int_{-1}^{y} P_X(x)dx.$$

So the contrast response function of the LMC's will be proportional to the cumulative distribution function of the environment's contrasts. This is indeed found to be the case as can be seen in figure 1.2 and showcases the application of information-theoretical methods in neuroscience.

It is often useful to rephrase this approach in terms of the redundancy of a code. Namely, defined the capacity of a response model $P_Y(y|X = x)$ as the maximum mutual information between $X$ and $Y$. More specifically,

$$C = \max_{P_X(x)} I(X;Y).$$

For discrete random variables, clearly the maximum of the mutual information is just the entropy of the stimulus $H[X]$, which is achieved by a deterministic one-to-one mapping from $X$ to $Y$. One can then define the redundancy of some response distribution $P_Y(y|X = x)$ as

$$\mathscr{R} = 1 - \frac{I(X;Y)}{C}.$$

The redundancy of a code measures how far its information transmission is from the optimum given by the capacity. That is, if a code has redundancy of 0, it is optimally coding for the stimulus $X$ in the responses $Y$. A redundancy of 1, on the other hand, means no information is being transmitted at all. Considering multiple neurons, the response can be written as $Y = (Y_1, \ldots, Y_N)^\top$, and one can write the redundancy as

$$\mathscr{R} = \frac{1}{C}\left(C - \sum_i H[Y_i]\right) + \frac{1}{C}\left(\sum_i H[Y_i] - H[Y]\right)$$

The first term accounts for the redundancy arising from unequally frequent usage of different responses from individual neurons and the second term accounts for the redundancy arising from correlations between the activities $Y_i$. In the example above we have only had to deal with the left term, since we had only one activity and therefore no correlations between them. An extensive body of efficient coding literature,[22] however, has dealt with the second term, and a number of different approaches have looked towards independent components of natural stimuli, assuming that whitening or gain control could account for the maximisation of the first term.

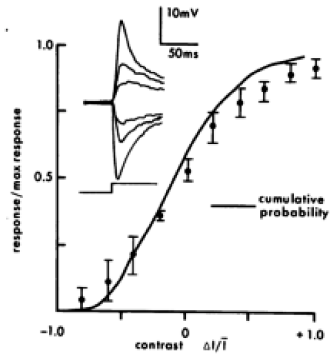


Figure 1.3: The response function of the blowfly LMC closely resembles the cumulative distribution of visual contrasts in its natural environment. Figure taken from (Laughlin, 1981)

[22] Bell, A. J. and Sejnowski, T. J. (1997). The independent components of natural scenes are edge filters. *Vision research*, 37(23):3327–3338; Lewicki, M. S. (2002). Efficient coding of natural sounds. *Nature neuroscience*, 5(4):356–63; and Hahnloser, R. H. R. and Bla, F. (2011). An Efficient Coding Hypothesis Links Sparsity and Selectivity of Neural Responses. *October*, 6(10)

*Fisher Information*

Another very popular way to quantify the information content of a neural code is the Fisher information. The Fisher information is a concept from frequentist statistics and is formulated in a slightly different framework than the mutual information. In this case, we will not consider the state of the environment to be a random variable but to be a fixed unknown value $x$. Suppose further we are given a set of observations $Y$ distributed according to $P_Y(y;x)$, where $x$ is now regarded as a parameter of the distribution rather than a conditioning variable. The Fisher information of $Y$ is then a function of $x$ given by

$$\mathscr{J}(x;Y) = \boldsymbol{E}_Y\left[\left(\frac{\partial \log P_Y(y;x)}{\partial x}\right)^2\right].$$

To better understand the significance of the Fisher information it is useful to formulate a simple estimation problem. Say I am trying to estimate the true value of $x$ from an observation of $Y$, $y$. The probability of the observations given the value of $x$ is $P_Y(y;x)$ but it is often more convenient to look at the log-likelihood $\log P_Y(y;x)$. The maximum likelihood estimator of the parameter $x$ is then given by

$$\hat{x}_{MLE} = \arg\max_x \log P_Y(y;x).$$

One can then investigate how sensitive the estimator of $x$ is, by looking look at how much the probability of the observed data would change with a change of $x$. Taylor expanding the log-probability around the maximum likelihood estimator, the first term will be zero, and the second term will be given by

$$(\hat{x}_{MLE} - x)^2 \frac{\partial^2 \log P_Y(y;\hat{x}_{MLE})}{\partial x^2}.$$

The Fisher information can be shown to be equal to

$$\mathscr{J}(x;Y) = -\boldsymbol{E}_Y\left[\frac{\partial^2 \log P_Y(y;x)}{\partial x^2}\right],$$

which is the average of the coefficient of the Taylor expansion above.[23] So the Fisher information quantifies how sensitive the maximum likelihood estimator of the parameter $x$ is on average, when the true value of the parameter is $x$. That is, if $\mathscr{J}(x;Y)$ is high, the probability decays very quickly around the estimate, yielding a sharp estimate of $x$. On the other hand, if $\mathscr{J}(x;Y)$ is low, the minimum will have a shallow curvature around it, meaning the ML estimate will be imprecise. This is a notion of discriminability, telling us how precisely a given neural code allows us to specify the value of $x$.

[23] The maximum likelihood estimator is a maximum, so the second derivative will be negative, hence the Fisher information is positive.

The Fisher information is also related to estimation by the Cramér-Rao bound. Given an unbiased estimator of the parameter $x, \hat{x}(y)$, the Cramér-Rao bound states that

$$\boldsymbol{E}_Y\left[(x-\hat{x}(y))^2\right] \leq \frac{1}{\mathscr{J}(x;Y)}$$

This is an important result in statistics, and it has gained popularity in the neuroscience literature due to the simple form the Fisher information takes in the case of Poisson rate models. I will discuss these issues further in chapter 3.

*Estimation and Mean-Squared-Error*

The Fisher information provides a lower bound for the mean squared error of any unbiased estimator of the environment's state $x$. From a Bayesian perspective, however, the optimal estimator which minimises the MSE is easily computable, and is equal to the posterior mean of the random variable $X$ conditioned on the observed neural responses $y$

$$\hat{X}(y) = \boldsymbol{E}\left[X \mid Y = y\right].$$

In the feedforward paradigm described above, populations of neurons are concerned with computing some interesting aspect of the environment from their noisy inputs. This is very similar to the estimation problem described here. It makes sense then, that the population would minimise the mean squared error of estimating that particular feature from the neural response, say the presence of a predator call in an auditory stimulus or the direction of motion of an object in the visual field. In principle one could argue that other loss measures would make more sense from a physiological point of view, but that does not change the conclusions of this line of reasoning drastically.

The MMSE-based approach is not quite as popular as the two previous frameworks, but it provides a number of advantages. Sadly, there are no simple relations as can be found for Poisson models and the Fisher information, but the MMSE approach is much more flexible. For one it is relatively straightforward to include the temporal dimension in this framework, without any fundamental changes to the theory. Furthermore, temporal estimation has had a lot of interest in the signal processing community, and a number of different techniques have emerged which can be leveraged to obtain quality measures of a code through the optimal Bayesian estimator and its MMSE.

### *1.3 Neural Decoding and Population Codes*

I have so far refrained from discussing the exact method by which the population of neurons responds to the stimulus present in the environment. There are many ways to describe a neuron's activity, ranging from complex biophysical compartmental models focusing on the physiological properties of the neuron to simplified probabilistic models, which focus on the computational properties of neurons and populations thereof. The most notable neuron model is probably the Hodgkin-Huxley model of the squid giant axon, which first shed light on the biophysical mechanisms leading to the generation of action potentials.[24] This model, however, describes the temporal dynamics of the membrane potential of the neuron as a function of the injected currents into the synapses of the neurons. This is a great description of the biophysical properties of a neuron, but to place it in a coding framework would force us to simulate the spiking activity of the population of neurons sending inputs to that particular cell. There is a whole spectrum of neuron models, from the compartmental Hodgkin-Huxley models, to integrate-and-fire models all the way to probabilistic models of spiking processes.[25] Though similar decoding approaches have been developed for more complex neuron models,[26] I will here focus on simpler models of spiking, which allow one to treat the probability of a spike being fired analytically.

[24] Hodgkin, A. L. and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500

[25] Gerstner, W. and Naud, R. (2009). How good are neuron models? *Science*, 326(5951):379–380

[26] Gerwinn, S., Macke, J., and Bethge, M. (2009). Bayesian Population Decoding of Spiking Neurons. *Frontiers in computational neuroscience*, 3(October):14

The usual framework to study neural coding experimentally at the level of single cells involves surgically inserting electrodes into the cortex and recording the activity of neuron's during some kind of experiment. We are mostly interested in the coding of sensory information, in which case the experiment is usually the presentation of some sensory stimulus, often coupled with a subsequent behavioural task for the experimental subject. For example, in (Benucci et al., 2009), the experiment consisted of measuring responses from the primary visual cortex of anaesthetised cats upon the presentation of moving gratings in a given direction. It is generally known that cells in V1 respond to this kind of stimulus, but how can one determine how well the stimulus is encoded in a neural response? One simple way is to try to decode the stimulus explicitly from the neural response and see how well one performs. This is the so-called neural decoding approach, where one plays the part of a downstream cortical area and tries to decode the information encoded by a given cortical area. One can then resort to any of a number of estimation methods to decode the stimulus encoded by the upstream area. I will consider a couple of examples here, such as the MMSE-optimal estimator, particle filters and assumed density filtering, but this list is by now means exhaustive. In

(Berens et al., 2012), for example, the authors used a simple logistic regression model to estimate the stimulus from the neural responses recorded.

*Rate Codes and Temporal Codes*

How exactly can we model the dependence of the activity of a neuron on the stimulus? One can treat the neuron's response as a random variable and look for a mapping from the relevant stimulus directly to the activity of the neuron. This forces one to think about which aspect of the neuron's activity is relevant to the brain's functioning. Often one characterises the neuron's response solely by the number of spikes it emits. This leads to a rate code, where a neuron's response is given simply by the rate of responses the stimulus elicits. One can then think of a rate function which specifies the rate of a particular neuron as a function of the stimulus presented to the organism. For example, consider a neuron in V1 which responds to moving gratings with a given direction $\theta$. The expected number of spikes fired by that particular neuron is given by some function

$$R^i(\theta, T) = \boldsymbol{E}\left[\#\text{of spikes fired by the neuron } i \text{ in response to stimulus } \theta \text{ in T seconds}\right],$$

where the expectation is over multiple presentations of the gratings moving in a direction $\theta$. This is often called the tuning function of the neuron. The tuning function does not fully characterise the neuron's activity though, as to perform the decoding of the stimulus from the neural response, one would need the full probability distribution[27]

$$P(\text{number of spikes}|\theta).$$

[27] One would also need a prior distribution over $\theta$, in principle, but in the case of moving gratings one could simply assume the distribution to be uniform.

I will denote the number of spikes fired by neuron $i$ up to time $t$ by $N^i(t)$. When the time of pooling is always the same throughout the analysis, i.e. when the response of the neuron is recorded for a specified time $T$ after the stimulus presentation, one can drop the time dependence and just write $N^i$. Therefore, if I record a set of responses $\{N^i\}$ from a subject's neurons and assume the stimulus is drawn from some distribution $P(\theta)$, the posterior distribution over the presented stimulus is given by

I will denote the full set of responses of all neurons by $\{N^i\}$.

$$P(\theta|\{N^i\}) = \frac{P(\{N^i\}|\theta)P(\theta)}{P(\{N^i\})}.$$

It is then simple to obtain the Bayesian estimator from the posterior distribution.

As mentioned, I will here concentrate on the Poisson distribution for the sake of modelling the probability of a spike

being fired. The Poisson distribution of a spike count conditioned on the presented stimulus is

$$P_{Poiss}(N^i|\theta) = \frac{e^{-R(\theta)}R(\theta)^{N^i}}{N^i!}.$$

The Poisson distribution describes random events which occur independently with a certain rate at every instant. If the duration of our experiment is $T$, one can write $r(\theta) = R^i(\theta,T)/T$, and the probability of observing a spike of neuron $i$ in a infinitesimal interval $dt$ would be given by $r^i(\theta)dt$, leading to

$$P_{Poiss}(N^i|\theta) = \frac{e^{-r^i(\theta)T}\left(r^i(\theta)T\right)^{N^i}}{N^i!}.$$

It is easy to show that this distribution can be obtained by writing the probability of a spike or the absence of a spike for every time interval $dt$. For the intervals where a spike occurred, the probability is $r^i(\theta)dt$ and for the intervals where no spike occurs it is $1-r^i(\theta)dt$. Multiplying the terms leads to the probability density of observing a set of spikes at a given ordered set of spike times $\{t_1,\ldots,t_{N^i}\}$

$$P(\{t_1,\ldots,t_{N^i}\}|\theta) = (r^i(\theta)dt)^{N^i}(1-r^i(\theta)dt)^{T/dt-N^i}.$$

Integrating over all possible values of the spike times, one obtains

$$P(N^i|\theta) = \frac{(r^i(\theta)T)^{N^i}}{N^i!}(1-r^i(\theta)dt)^{T/dt-N^i}.$$

Now taking the limit $dt \to 0$ leads to

$$P_{Poiss}(N^i|\theta) = \frac{e^{-r^i(\theta)T}\left(r^i(\theta)T\right)^{N^i}}{N^i!}.$$

One additional assumption which is often made in the neural coding literature is the conditional independence of the neuron's firing given the presented stimulus. This means that

$$P(\{N^i\}|\theta) = \prod_i P(N^i|\theta).$$

With those hypotheses it is easy to formulate a full decoding framework for a given experiment. One still needs to experimentally estimate the tuning functions $R^i$, and there are a number of different tools for that. The most obvious choice would be to present each stimulus repeatedly and take the average number of spikes as the rate, but there are many ways to improve on that.[28]

[28] For a more complete review see (Dayan and Abbott, 2001). For an example of more recent techniques for tuning function and receptive field estimation see (Park and Pillow, 2011).

*Dynamic Population Coding*

The rate coding framework is very convenient, as it allows one to simply estimate the tuning functions and makes decoding very simple, but it places a very restrictive assumption on the nature of the stimulus. More precisely, if one only measures the response as the rate of the neuron's spiking, one needs to pool the responses for a certain time, and any changes the stimulus undergoes in a timescale smaller than the pooling time will be completely lost. An alternative to considering only the spike counts in a given time interval is to consider the whole time-dependent spike train as the neural response. For that one must model the full dependence of the spike times $\{t_k^i\}$ of the $k$-th spike of every neuron $i$ on the stimulus $X$. Furthermore, the finding that high-level decisions are often made in less than 150 ms shows that information processing in the brain is possible at timescales that would render pooling of many spikes problematic.[29]

[29] Thorpe, S., Fize, D., and Marlot, C. (1996). Speed of processing in the human visual system. *Nature*, 381(6):520–522

One can then consider how to model the full spike train probability as a function of the stimulus. Furthermore, it is not needed to assume the stimulus to be static throughout the experiment anymore. I will denote by $N_{0:t}^i$ the spike count of neuron $i$ at every time $0 < s < t$. This formalises the notion of the spike train of a neuron. An alternative description would be the spike times $\{t_k^i\}$ for each neuron $i$. Say that the direction of the moving grating is now dynamic, given by $\theta(t)$. Assuming Poisson statistics, the probability density of observing a given spike train from neuron $i$ given the history of $\theta(t)$[30]

$$P\left(\{t_k^i\}|\theta_{0:t}\right) = \exp\left[-\int_0^t r^i\left(\theta(s)\right)ds\right]\prod_k r^i\left(\theta(t_k^i)\right). \tag{1.1}$$

[30] Note the absence of the term $1/k!$ in this expression. This is due to the fact that we are here defining a distribution over spike times. To recover the Poisson distribution, one would need to integrate over all possible values of $\{t_k^i\}$. Since they are ordered, integration is cumbersome. That can be solved by integrating over all spike times and then dividing by all possible orderings to compensate for that, yielding the previous expression for the Poisson distribution.

The process $N^i(t)$ is usually called an inhomogeneous Poisson process, as the rate depends on the time. Furthermore, if $\theta$ is itself a random variable, the resulting point process is called a doubly stochastic Poisson process, since the rate is also a random variable.

This approach can be further extended by allowing the rate $r$ to depend on the history of the stimulus and on the history of the spiking process itself. One way to do so is with Generalised Linear Models (GLM's) which model the rate as the exponential of a linear function of the stimulus history and of the spiking history. These models have become very popular in the computational neuroscience literature, as they allow to model fairly complex neural responses and a wide range of spiking behaviours.[31] GLM's are too complex to allow for a general analytic treating, however, so I will not focus on them. I will, however, consider a model of adaptation that

[31] A notable application of GLM's in neural coding was (Pillow et al., 2008), where a complete neuronal population had its activity recorded, modelled and decoded by GLM's. For a review of the decoding problem with a focus on GLM's see (Ahmadian et al., 2011; Pillow et al., 2011).

allows one to model a simple kind of history dependence in the spiking process $N(t)$.

In this setting, one could then constrain the type of tuning functions $r$ to belong to a family of functions and ask which of these tuning functions gives the best performance when reconstructing the stimulus. This would give one the best encoder in the family of functions with respect to a reconstruction task. A notable example from the literature is the work of Zhang and Sejnowski,[32] where the authors followed this reasoning, using the Fisher information as a measure of the encoder's performance. There they concluded that for a general family of radial tuning functions of the form

$$r(x) = \phi k\left(\frac{|x-c|^2}{\alpha^2}\right),$$

the Fisher information could be written as

$$\mathscr{J} = \eta \alpha^{D-2} K_k(\phi, T, D),$$

where $\eta$ is a measure of the density of packing of the neurons, $D$ is the dimension of the stimulus space, $T$ is the duration of the experiment. and $K_k$ is a constant which depends on the specifics of the function $k$.

Surprisingly, this tells one that the dependence of the Fisher information on the width of the tuning functions is independent of the specific shape of the tuning functions $r$, which only contribute through a normalisation factor $K_k(\phi, T, D)$. Furthermore, it can also be extended to populations of neurons with different maximal firing rates $\phi$, leading to a similar result. Interestingly, this tells one that the Fisher information depends on the tuning width in a remarkably simple way. If one wants to maximise $\mathscr{J}$ as a function of $\alpha$ (therefore minimising the Cramér-Rao bound) one needs only to look at the dimension of the stimulus space. For $D = 1$, the optimal tuning width would be 0, leading to a vanishing error. For $D = 2$ the tuning width has no effect on the Fisher information, and for $D > 2$ broader tuning widths are always better. This behaviour of the optimal tuning width clearly poses some unsettling conclusions. First of all, the fact that the conclusion depends critically on the dimension of the stimulus space is somewhat curious. Second, this conflicts with ecological theories of sensory processing, which state that the response functions of sensory neurons are adapted to the statistics of the stimuli they respond to.[33] The formulation above leaves no room for an influence of the environment on the shape of the tuning functions, their optimal shape being dictated solely by the dimension of the stimulus space.

These and other shortcomings of the Fisher information as a measure of efficiency of a neural code have been addressed

[32] Zhang, K. and Sejnowski, T. T. J. (1999). Neuronal tuning: To sharpen or broaden? *Neural Computation*, 11(1):75–84

[33] Atick, J. J. (1992). Could information theory provide an ecological theory of sensory processing? *Network: Computation in neural systems*, 3(2):213–251

before in the literature,[34] and I will not spend much time discussing the Fisher information approach. I will focus most of the discussion in this thesis on the Bayesian estimation approach, as it allows one to account for the effect of the dynamical structure of the system and its effects on the optimal neural encoder.

[34] Bethge, M., Rotermund, D., and Pawelzik, K. R. (2002). Optimal short-term population coding: When Fisher information fails. *Neural Computation*, 14(10):2317–2351; and Yaeli, S. and Meir, R. (2010). Error-based analysis of optimal tuning functions explains phenomena observed in sensory neurons. *Frontiers in computational neuroscience*, 4(October):16

### *1.4 Neural Implementations*

One important aspect which I have not touched upon is the implementation in neural circuits of the computations discussed here. The posterior mean estimator gives the optimal reconstruction achievable from a certain type of observations, but the implementation of this reconstruction in a neural circuit is a completely different problem. In a setting very similar to the one I will consider in this work, (Bobrowski et al., 2009) have presented a simple spiking neural network which implemented the estimation of the posterior distribution in a simple way. The computations that can be implemented in a population of neurons have been a central topic in computational neuroscience, and one can argue that much of the field of artificial neural networks is concerned with analogous questions.

A very popular coding framework is the so-called probabilistic population coding framework (PPC) proposed in (Ma et al., 2006). In this formulation, the tuning functions are given by exponential distributions, and each spike contributes to the posterior log-likelihood with the addition of a new term, allowing for linear decoding of the posterior. This is somewhat similar to the case considering here, but the PPC formalism does not provide a simple way to include temporal stimuli. A lot of new contributions have been made recently to the area of neural coding (see (Boerlin and Denève, 2011) and (Beck et al., 2011) for example), and it is still an active area of research. I will not dive deeper into these aspects of coding, however, as I am mostly concerned with the encoding step and regardless of the neural implementation of the computation, the expected MSE of the posterior mean estimator is the minimum achievable with the information observed.

One weakness of the MSE approach which must be noted is the feedforward assumption. The Bayesian estimator is optimal for reconstructing the stimulus from the spikes emitted by some neural population. Neural populations, however, rarely work in this straight feedforward fashion and feedback modulation complicates the analysis a lot. One simple example would be a decoder which signals how confident it is in its estimate through feedback connections, allowing the sensory neurons to allocate its firing to areas with lower certainty

about the stimulus, or sharpening the tuning functions near the estimated value of the stimulus. There is no reason these kind of decoding mechanisms could not outperform a feedforward Bayesian estimator. The analysis of such feedback codes, however, is much more complicated, as the distribution of observations would now be dependent on the state of the estimator, and I will refrain from discussing this case in the present work.

*Structure*

THE MAIN GOAL OF THIS THESIS is to develop a conceptual framework for studying optimal population coding in a dynamic setting. I believe that the inclusion of time into the coding framework raises a number of questions, which have not been addressed in the scientific literature properly. In chapter 2 I will introduce the general theory of filtering of stochastic stimuli, giving special attention to the filtering of stochastic processes observed through doubly stochastic Poisson processes. After that, in chapter 3 I will discuss results regarding the Mean-Squared-Error (MSE) of optimal filters of point process observations, presenting a number of new analytic results. In chapter 4, I will generalise the filtering framework to control problems, showing results for optimal control theory of point process-observed processes. In chapter 5 I will then provide the connection to neuroscience, by considering the optimal encoding strategy for a population of neurons coding for a stochastic stimulus. I will then finalize by discussing the impact of the work presented and suggesting future research directions.

*Contribution*

THE MAIN CONTRIBUTION OF THIS THESIS is in providing a conceptual toolbox to study optimal coding problems in a dynamic environment. I propose that the study of the average performance of an optimal Bayesian filter reconstructing the relevant stimulus provides a good measure of the quality of a dynamic code. Using this framework, I derive analytic results for the fast population code for dense populations of Poisson neurons with Gaussian tuning functions.[35] These are to my best knowledge the first results of this kind obtained for neural coding of dynamic stimuli.

The results presented in this thesis have been published and presented throughout the duration of my doctoral studies. The findings in chapter 3 were first published at the *Neural Information Processing Systems* conference, where it was

[35] Huys, Q. J. M., Zemel, R. S., Natarajan, R., and Dayan, P. (2007). Fast population coding. *Neural Computation*, 19(2):404–441

presented as a poster in addition to the publication in the conference proceedings.[36] These results were then further developed and put in the greater context of computational neuroscience and published in a special edition of the *Journal of Statistical Mechanics: Theory and Experiment* title *Statistical Physics and Neuroscience*, which focused on the challenges neuroscience presented to statistical physics.[37] The results presented chapter 4 have been submitted to the *NIPS* conference proceedings as well, and are currently under review.[38]

Parallel to the topics presented here I have also contributed to other ongoing research projects during my doctoral studies. In a research project headed by fellow doctoral student Chris Häusler and myself, we have proposed a novel way of training temporal Boltzmann machines, which improves their performance as generative models of temporal data greatly. This was presented in a workshop on Deep Learning at the *NIPS* conference as well.[39] This was then used as a model for temporal sparsity in visual cortex and results on sequences of natural images were published in the journal *Brain Research* in a special issue on neural coding.[40] The advantages of the training procedure for generative models of temporal data as well as for forecasting were further extended on and are currently under review for publication in the journal *Neurocomputing*.[41]

In addition to these projects, I have also worked on the publication of a manuscript originating from my Masters thesis, which was since published in the journal *Physica A*. There we investigated the effect of different learning strategies on the emergence of moral opinions in a model of social learning.[42]

[36] Susemihl, A., Meir, R., and Opper, M. (2011). Analytical Results for the Error in Filtering of Gaussian Processes. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P., Pereira, F., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, pages 2303–2311

[37] Susemihl, A., Meir, R., and Opper, M. (2013). Dynamic state estimation based on poisson spike trains: towards a theory of optimal encoding. *Journal of Statistical Mechanics: Theory and Experiment*, 2013(03):P03009

[38] Susemihl, A., Meir, R., and Opper, M. (2014). Optimal Population Codes for Control and Estimation. *ArXiv e-prints*

[39] Häusler, C. and Susemihl, A. (2012). Temporal autoencoding restricted boltzmann machine. *arXiv preprint arXiv:1210.8353*

[40] Häusler, C., Susemihl, A., and Nawrot, M. P. (2013a). Natural image sequences constrain dynamic receptive fields and imply a sparse code. *Brain research*, 1536:53–67

[41] Häusler, C., Susemihl, A., Nawrot, M. P., and Opper, M. (2013b). Temporal autoencoding improves generative models of time series. *arXiv preprint arXiv:1309.3103*

[42] Vicente, R., Susemihl, A., Jericó, J., and Caticha, N. (2014). Moral foundations in an interacting neural networks society: A statistical mechanics analysis. *Physica A: Statistical Mechanics and its Applications*

# 2
# *Filtering and Prediction with Point Process Observations*

> Prediction is very difficult,
> especially about the future.
>
> Niels Bohr

In the Introduction I described the general framework in which I seek to study optimal population coding. To do so in a dynamic setting, one must first develop the theory of temporal estimation of dynamic stimuli from point processes. This is a surrogate for the functioning of a neural population receiving information from the encoder. There are a number of cases in which the filtering problem can be solved exactly, and I will discuss results from the theory of optimal filtering. In the cases where the optimal filter is intractable or too expensive to evaluate exactly, I will present methods to approximate the posterior density.

In the neuroscientific context, the process $X(t)$ being estimated or filtered would correspond to some environmental feature of interest to a sensory system of some organism, while the signal would be some neural response coming from a neural population. As was mentioned in section 1.3, one example would be to estimate the presence of a moving grating from the response of retinal ganglion cells. In that sense, the ganglion cells provide a noisy representation of an environmental variable of interest to some downstream cortical area (V1 for example). In this chapter I will discuss methods to infer the value of the sensory stimulus from the noisy response of a population of neurons.

## *2.1 A Note on Stochastic Processes*

Throughout this thesis, I will repeatedly talk about stochastic processes and the framework of stochastic calculus, so I will provide a short introduction to stochastic processes and the theory of stochastic calculus. In the study of ordinary differ-

ential equations, one works with equations such as

$$\frac{dx}{dt} = f(x),$$

which are solved by

$$x(t) = x(0) + \int_0^t f(x(u),u)du.$$

This can be modified to include a white-noise term in the evolution of $x(t)$, leading to the Langevin equation

$$\frac{dX}{dt} = f(X,t) + \sigma(X,t)\xi(t),$$

where $\sigma(X(u),u)$ is a state- and time-dependent strength and $\xi(t)$ is a *rapidly fluctuating random term*, i.e.[1]

$$\boldsymbol{E}[\xi(t)] = 0 \text{ and } \boldsymbol{E}[\xi(t)\xi(s)] = \delta(t-s).$$

This approach is problematic, however, since the process $X(t)$ thus defined is not differentiable, rendering the Langevin equation mathematically inconsistent. One can, however, extend the solution to the deterministic case as [2]

$$X(t) = X(0) + \int_0^t f(X(u),u)du + \int_0^t \sigma(X(u),u)\xi(u)du.$$

$\xi(u)du$ can be shown to be equal to $dW(u) = W(u+dt) - W(u)$ in the limit $dt \to 0$, which leads to the usual Itō stochastic integral

$$X(t) = X(0) + \int_0^t f\left(X(u),u\right)du + \int_0^t \sigma\left(X(u),u\right)dW(u).$$

The stochastic integral can be shown to exist as long as the functions $f$ and $\sigma$ are continuous and non-anticipating.[3] One usually writes the evolution of $X(s)$ in terms of a stochastic differential equation, instead of a stochastic integral. The process $X(t)$ described above would obey the SDE

$$dX(t) = f\left(X(t),t\right)dt + \sigma\left(X(t),t\right)dW(t).$$

This is just a shorthand for the stochastic integral, and has no precise mathematical interpretation, as the terms are of different orders. More specifically, the term $dW(t)$ is of the order of $\sqrt{dt}$ while the first term is of order $dt$. In an analogy to the study of classical mechanics, the first term is often called the drift of $X(t)$ and the second the diffusion.

Processes $X(s)$ defined in this way are continuous. Often, however, one wants to model a stochastic process which incurs discontinuous jumps as well. I can for that purpose introduce a third term in the definition of $X(t)$. Let $j(X(t),t)$

[1] Gardiner, C. W. (2004). *Handbook of Stochastic Methods: for Physics, Chemistry and the Natural Sciences*, volume Vol. 13 of *Series in synergetics*. Springer

[2] Note that because of the new definition, $X(t)$ is now a random variable, hence the upper-case notation.

[3] A function $G(t)$ is said to be non-anticipating with respect to a stochastic process $X$ if it is statistically independent of values of $X(s)$ for $s > t$. Simply put, $G_n(t) = \int_0^t dW(s)$ is non-anticipating with respect to $W(s)$, but $G_a(t) = \int_0^{2t} dW(s)$ is not.

be a function that describes the size of the jump the process experiences at time $t$ and state $X(t)$. If these jumps occur at some set of random times $\{t_i\}$, the process $X(t)$ can be written as

$$X(t) = X(0) + \int_0^t f\left(X(u), u\right) du + \int_0^t \sigma\left(X(u), u\right) dW(u) + \sum_{t_i < t} j\left(X(t_i^-), t_i\right),$$

where I have defined

$$X(t^-) \equiv \lim_{s \uparrow t} X(s),$$

as the limit of $X(t)$ from the left. The jumps in $X(t)$ should be modulated by the point where they originate, not their destination, so this definition makes intuitive sense. Let $N(t)$ be then given by

$$N(t) = \sum_{t_i} \Theta(t - t_i) = \int_0^t \sum_i \delta(u - t_i) du \equiv \int_0^t dN(u),$$

where $\Theta(x)$ is the Heaviside step function. With this definition, I can then write

$$X(t) = X(0) + \int_0^t f\left(X(u), u\right) du + \int_0^t \sigma\left(X(u), u\right) dW(u) + \int_0^t j\left(X(u), u\right) dN(u).$$

Throughout the text I will also employ an SDE notation for this integral as follows

$$dX(t) = f\left(X(t), t\right) dt + \sigma\left(X(t), t\right) dW(t) + j\left(X(t^-), t\right) dN(t). \tag{2.1}$$

This encompasses all the stochastic processes I will consider in this text. What happens to a function of a stochastic variable that changes over time, though? Say I want to evaluate some function of $X(t)$, say $g(X(t), t)$, how does this function vary in time? Itō's lemma tells one how to find the variation in $g$ from the process $X(t)$. If $X(t)$ evolves according to equation (2.1), we have[4]

[4] For a more full derivation, see (Sennewald and Wälde, 2006) or (Privault, 2014).

$$\begin{aligned} dg &\equiv \lim_{dt \to 0} \left[g(X(t + dt), t + dt) - g(X(t), t)\right] \\ &= \left(\partial_t g + \partial_x g^\top f(X(t), t) + \frac{1}{2} \mathrm{Tr}\left[\sigma \sigma^\top \partial_x^2 g\right]\right) dt + \partial_x f^\top \sigma(X(t), t) dW(t) \\ &\quad + \left(g(X(t^-) + j(X(t^-, t), t) - g(X(t^-), t)\right) dN(t), \end{aligned}$$

where I am using the notation

$$\partial_x g = \left(\frac{\partial g}{\partial x_1}, \ldots, \frac{\partial g}{\partial x_N}\right)^\top$$

and

$$(\partial_x^2 g)_{i,j} = \frac{\partial^2 g}{\partial x_i \partial x_j}.$$

This is, again to be understood as a stochastic integral, where we have

$$g(X(t),t) = g(X(0),0) + \int_0^t \left( \partial_t g + \partial_x g^\top f(X(u),u) + \frac{1}{2}\mathrm{Tr}\left[\sigma\sigma^\top \partial_x^2 g\right] \right) du + \int_0^t \partial_x f^\top \sigma(X(u),u) dW(u) + \int_0^t \left(g(X(u^-) + j(X(u^-,u),u) - g(X(u^-),u)\right) dN(u).$$

This is in stark contrast of the usual change of variables formula for differentiable variables $y(t)$, where we would have

$$dg \equiv \lim_{dt\to 0} \left[g(y(t+dt),t+dt) - g(y(t),t)\right] = \left(\partial_t g + \partial_y g^\top \partial_t y\right) dt.$$

*The Evolution of Probabilities*

Another important question, is how $X(t)$ is distributed at some time $t$ if it is initially at some point $x_0$ at time 0.[5] It is useful for that problem to consider the transition probability density

[5] Or distributed according to some distribution $P_0(X)$ at time 0.

$$P(X(t+dt) \in A | X(t)) = \int_A p(x, t+dt | X(t), t).$$

Let me define three sources of change arising from the transition density $p$. For any $\varepsilon > 0$ I will assume the limits below exist

$$\lim_{dt\to 0} p(x,t+dt|z,t)/dt = W(x|z,t), \forall x,z,t, \text{ s.t } |x-z| \geq \varepsilon, \tag{2.2a}$$

$$\lim_{dt\to 0} \frac{1}{dt} \int_{|x-z|\leq\varepsilon} dx (x-z) p(x,t+dt|z,t) = A(z,t) + O(\varepsilon), \tag{2.2b}$$

$$\lim_{dt\to 0} \frac{1}{dt} \int_{|x-z|\leq\varepsilon} dx (x-z)(x-z)^\top p(x,t+dt|z,t) = B(z,t) + O(\varepsilon). \tag{2.2c}$$

These terms define the contribution of jumps ($W(x|z,t)$), drift ($A(z,t)$ and diffusion ($B(z,t)$) to the transition density of the process $X(t)$. It is straightforward to show that, if $X(t+dt)$ is given by $X(t) + dX(t)$ as in equation (2.1), then $A(z,t) = f(z,t)$ and $B(z,t) = \sigma(z,t)\sigma(z,t)^\top$. I have not defined the distribution of $dN(t)$, but assuming $N(t)$ is a Poisson process with rate $\lambda$, the jump term will be simply

$$W(x|z,t) = \lambda \delta(x - z + j(z,t)).$$

With these definitions in hand, it can be shown that the probability density $p$ will evolve according to the differential Chapman-Kolmogorov equation

$$\frac{\partial p(x,t|x_0,0)}{\partial t} = -\nabla\cdot(A(x,t)p(x,t|x_0,0)) + \frac{1}{2}\sum_{i,j}\frac{\partial^2}{\partial x_i \partial x_j}\left[B_{ij}(x,t)p(x,t|x_0,0)\right] + \int dz\,[W(x|z,t)p(z,t|x_0,0) - W(z|x,t)p(x,t|x_0,0)] \tag{2.3}$$

The first line corresponds to the terms found in the Fokker-Planck equation, while the second line corresponds to the terms found in the Master equation. These equations are usually used to describe drift-diffusion and pure jump processes respectively. The differential Chapman-Kolmogorov equation generalises both equations to processes with drift, diffusion and jumps.[6]

[6] For a full account of the differential Chapman-Kolmogorov equation, see (Gardiner, 2004).

*Smooth Markovian Processes*

The stimuli defined by SDE's like equation (2.1) will often yield sample paths which are not differentiable. I am, however, interested in using the theory of stochastic processes to describe the natural stimuli a sensory system encounters in its environment, so it makes sense to consider smooth, differentiable processes as well. Huys et al. (2007) looked at a number of Gaussian processes which yield smooth sample paths. I will here consider a type of process which I shall call the Matern process throughout this thesis.[7] Symbolically, one can write these processes as

$$\left(\frac{d}{dt}+\gamma\right)^P X(t) = \eta\frac{dW(t)}{dt},$$

[7] I will call these processes Matern processes because their autocorrelation $k(t,u) = \boldsymbol{E}[X(t)X(u)]$ are given by the Matern kernel described in (Rasmussen and Williams, 2005).

where $P$ is the order of the process. Clearly, this notation is not precise, since the Wiener process $W(t)$ is not differentiable. This can be written as a system of SDE's as

$$\dot{X}_1(t) = X_2(t), \qquad \ldots, \qquad \dot{X}_{P-1}(t) = X_P(t), \qquad dX_P(t) = -\sum_{i=1}^{P}\gamma^{P+1-i}X_i\,dt + \eta\,dW(t).$$

If $P = 1$, this gives the one-dimensional Ornstein-Uhlenbeck process. If I take $P > 1$, however, $X_1(t)$ will be a smooth random process, as can be seen in figure 5.2. $X_1(t)$ itself is no longer a Markov process, as its evolution depends on its time derivatives as well as of its state. It is, however, possible to embed the process $X_1(t)$ in a $P$-dimensional space, along with its $P-1$ first derivatives, rendering it Markov again. In (Susemihl et al., 2013) I have used this to study the MMSE

of smooth processes. In this way, all the tools of stochastic dynamics are still available, but one can consider smooth processes, more similar to the ones observed in nature.

When studying a population of neurons responding to such an embedded smooth process $X(t) = (X_1(t), \ldots, X_P(t))^\top$, I will mostly consider tuning functions which only depend on the original smooth process given by $X_1(t)$, which leads to the same filtering process considered in (Huys et al., 2007).

*Infinitesimal Generator of a Stochastic Process*

The infinitesimal generator of a stochastic process is defined as the operator

$$\mathscr{A} f(x) = \lim_{dt \to 0} \frac{\boldsymbol{E}\left[f(X(t+dt)) | X(t) = x\right] - f(x)}{dt}.$$

The adjoint of this operator is defined as the operator $\mathscr{A}^\dagger$ satisfying

$$\int (\mathscr{A} f(x)) g(x) dx = \int f(x) (\mathscr{A}^\dagger g(x)) dx.$$

## 2.2 Estimation and Filtering

Estimation is the field of statistics that deals with the inference of some unknown variable from uncertain observations of that variable. It can be best described by an example. Given a pair of variables $X$ and $Y$, and a model for their relationship, say $P_Y(y|X = x)$, one could infer the value of $X$ from observations of $Y$. Using Bayes' rule one obtains

$$P_X(x|Y = y) = \frac{P_X(x) P_Y(y|X = x)}{P_Y(y)},$$

which can be used to estimate the value of $X$. I will be mostly concerned with temporal processes, say a random process $X(t)$ which needs to be inferred from observations of a dependent process $Y(t)$. When one is interested in inferring $X(t)$ from data $\{Y(s)\}$, $s \in [0, T]$, the problem gets named according to the value of $t$. If $t \in [0, T]$, it is called a *smoothing* problem. If $t = T$, it is called a *filtering* problem. If $t > T$, it is called a *prediction* or *forecasting* problem. The temporal structure of the processes leads to correlations in the variables being estimated ($X(s)$ for different values of $s$), and there a number of ways to take advantage of this. I will look into the theory of filtering of diffusion processes observed through a second diffusion process dependent on the first and then turn to the theory of filtering of diffusion processes observed through doubly stochastic point processes.

Smoothing, filtering and predicting.

*Kalman Filtering*

Let me consider a more concrete setting. Suppose one is dealing with a system that evolves according to a stochastic discrete-time dynamics given by

$$X(t+1) = AX(t) + H^{1/2}N_t.$$[8]

We take $X(t) \in \mathbf{R}^n$, $A \in \mathbf{R}^{n*n}$ and $H \in \mathbf{R}^{n*n}$ positive-definite. $N_t$ is a normal n-dimensional random variable with zero mean and unit standard deviation. Suppose now we observe a process $Y(t)$ given by

$$Y(t) = CX(t) + D^{1/2}M_t,$$

where $C \in \mathbf{R}^{m*n}$, $Y(t) \in \mathbf{R}^m$ and $D \in \mathbf{R}^{m*m}$ positive-definite. $M_t$ is as before a normal $m$-dimensional random variable with zero mean and unit standard deviation. The filtering problem is to determine an estimate of $X(t)$ given observations of $Y(1), Y(2), \ldots, Y(t)$. This can done by a recursive estimation procedure first proposed by Rudolf E. Kálmán. Namely, for each time step, one first predicts the conditional distribution of $X(t)$ given our estimate of $X(t-1)$ and then corrects that according to the observation $Y(t)$. One can easily obtain recurrence relations for this filtering problem by noting how the mean and variance of $X(t)$ evolve. One has

$$\boldsymbol{E}\left[X(t+1)\middle|\mu(t),\Sigma(t)\right] = A\boldsymbol{E}\left[X(t)\right],$$

and

$$\boldsymbol{E}\left[X(t+1)X(t+1)^\top\middle|\mu(t),\Sigma(t)\right] = A\boldsymbol{E}\left[X(t)X(t)^\top\right]A^\top + H,$$

which leads clearly to

$$\boldsymbol{E}\left[X(t+1)X(t+1)^\top\middle|\mu(t),\Sigma(t)\right] - \boldsymbol{E}\left[X(t+1)\middle|\mu(t),\Sigma(t)\right]\boldsymbol{E}\left[X(t+1)\middle|\mu(t),\Sigma(t)\right]^\top = A\Sigma(t)A^\top + H.$$

So, in the absence observations, if knowledge of $X(t)$ was given by $\mathcal{N}(\mu(t),\Sigma(t))$, the distribution over $X(t+1)$ before the observations is[9]

$$\mathcal{N}(A\mu(t), A\Sigma(t)A^\top + H).$$

After observing the value of $Y(t+1)$, one can update the distribution through Bayes' rule as

$$P(X(t+1)|Y(t+1),\mu(t),\Sigma(t)) = \frac{P(Y(t+1)|X(t+1))P(X(t+1)|\mu(t),\Sigma(t))}{P(Y(t+1)|\mu(t),\Sigma(t))}.$$

Here I have dropped the verbose notation of $P_{X(t+1)}(x|Y(t+1) = y;\mu(t),\Sigma(t))$, and have written that simply as $P(X(t+1)|Y(t+1),\mu(t),\Sigma(t))$. The meaning should be clear from

[8] $H^{1/2}$ indicates the Cholesky decomposition of the positive-definite (or semi-definite) matrix $H$. The exponent $1/2$ is used because $H^{1/2}\left(H^{1/2}\right)^\top = H$.

[9] $\mathcal{N}(\mu,\Sigma)$ denotes the normal probability density function with mean $\mu$ and covariance $\Sigma$. The density function is given by

$$\mathcal{N}(\mu,\Sigma) = \frac{1}{(2\pi)^{N/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^\top\Sigma^{-1}(x-\mu)},$$

where $N$ is the dimension of $x$, and $|\Sigma|$ is the determinant of the covariance matrix $\Sigma$.

the context. Furthermore, the distribution $P(X(t+1)|\mu(t),\Sigma(t))$ is given by the Chapman-Kolmogorov equation as

$$P_{X(t+1)}(x|\mu(t),\Sigma(t)) = \int dz P_{X(t+1)}(x|X(t)=z)P_{X(t)}(z|\mu(t),\Sigma(t)),$$

leading to the relations derived above. Note that both terms in the numerator of the Bayesian update are Gaussian distributions and the denominator does not depend on $X(t+1)$, so one can simply find the mean and covariance by looking at the exponents in the numerator. The log probabilities are

$$\log\left[P(Y(t+1)|X(t+1))\right] = -\frac{1}{2}(Y(t+1)-CX(t+1))^\top D^{-1}(Y(t+1)-CX(t+1))$$

disregarding the normalization factor, and

$$\log\left[P(X(t+1)|\mu(t),\Sigma(t))\right] = -\frac{1}{2}(X(t+1)-A\mu(t))^\top (A\Sigma(t)A^\top+H)^{-1}(X(t+1)-A\mu(t))$$

disregarding the normalization factor. Collecting terms one obtains

$$\log\left[P(X(t+1)|Y(t+1),\mu(t),\Sigma(t))\right] = -\frac{1}{2}(X(t+1)-\mu(t+1))^\top \Sigma(t+1)^{-1}(X(t+1)-\mu(t+1)),$$

again disregarding the normalization factor, where

$$\Sigma(t+1) = \left(\left(A\Sigma(t)A^\top+H\right)^{-1}+C^\top D^{-1}C\right)^{-1}$$

$$\mu(t+1) = A\mu(t)+\Sigma(t+1)C^\top D^{-1}(Y(t)-CA\mu(t)).$$

This formulation leads to somewhat cluttered recurrence relations. In the theory of Kalman filtering these are usually broken down into subsequent prediction and correction steps. The notation usually employed in filtering theory is to write $\mu_{t|t-1}$ and $\Sigma_{t|t-1}$ for the mean and covariance of the distribution $P(X(t)|\mu(t-1),\Sigma(t-1))$,[10] and $\mu_{t|t}$ and $\Sigma_{t|t}$ for the mean and covariance of the updated distribution $P(X(t)|Y(t))$.[11] One can write simply

[10] the prediction step

[11] the correction step

$$\begin{aligned}\mu_{t+1|t} &= A\mu_{t|t},\\ \Sigma_{t+1|t} &= A\Sigma_{t|t}A^\top+H.\end{aligned}$$

Defining the innovation term $Z(t+1)$, and its covariance by

$$\begin{aligned}Z_{t+1} &= Y(t+1)-C\mu_{t+1|t}\\ S_{t+1} &= C\Sigma_{t+1|t}C^\top+D.\end{aligned}$$

The *optimal Kalman gain* will be

$$K_{t+1} = \Sigma_{t+1|t}C^\top S_{t+1}^{-1}$$

The term *optimal Kalman gain* is usually employed in filtering theory, as it is the matrix $K$ that gives the minimum variance unbiased estimator of $X(t)$ given $Y(t)$.

and the posterior mean and covariance can be written as

$$\begin{aligned}\mu_{t+1|t+1} &= \mu_{t+1|t}+K_{t+1}Z(t+1)\\ \Sigma_{t+1|t+1} &= (I-K_{t+1}C)\,\Sigma_{t+1|t}\end{aligned}$$

It is relatively simple to show that these relations are equivalent to the ones derived above.

The Kalman filter is a fundamental tool in engineering and signal processing and has been used in anything from radar signal analysis to computer vision tracking and space expeditions. The list of applications is enormous, and I will only mention three examples. One application is to use the Kalman filter to estimate the current position of an object in a navigation system (see (Brown, 1973)). Another interesting application is the monitoring of positional measurements through a radar. The nature of radar measurements lends itself nicely to this formalism and the Kalman filter has been used extensively in these kinds of applications (see (Pearson and Stear, 1974)). These are classical examples, but the applicability of the Kalman filter is very widespread, and one can find examples of applications in unexpected fields, such as the estimation of future retail sales (see (Conrad and Corrado, 1979)).

It has a number of limitations, though. First, note that it requires the knowledge of the matrices governing the system's dynamics. If the system is governed by linear dynamic and the matrices are known, the Kalman filter provides the exact posterior probability. If these are unknown, however, one is forced to estimate them from data, and in the mismatched case the Kalman filter is an approximate method, and can lead to poor results. A number of extensions to the Kalman filter exist, such as the extended Kalman filter, the unscented Kalman filter and others. More recently sequential Monte Carlo Markov Chain methods known as particle filters have been a subject of great interest as they overcome a number of limitations of Kalman filters, mainly through sampling from many hypothetical system paths for $X$ and reweighing them to account for the observations.[12]

The mismatched case refers to the situation where the parameters of the system are unknown, and we are forced to use a model with parameters mismatched to the system's parameters.

[12] Doucet, A., De Freitas, N., and Gordon, N. (2001). An introduction to sequential monte carlo methods. In *Sequential Monte Carlo methods in practice*, pages 3–14. Springer

*Continuous Time: The Kalman-Bucy Filter*

The Kalman filter deals with discrete time systems and can be easily extended to continuous time systems. Though it can be rigorously proved that the derived filter equations are rigorous using stochastic calculus, I will only provide an informal derivation.Consider a linear stochastic differential equation, say

$$dX(t) = AX(t)dt + H^{1/2}dW(t), \tag{2.4}$$

where $W(t)$ is a Wiener process. Note that the correspondence with the discrete time case can be simply made by taking $A' = I - Adt$ and $H'^{1/2} = \sqrt{dt}H^{1/2}$. The obvious extension for the observation process to continuous-time would be

$$Y(t) = CX(t) + N(t),$$

where $N(t)$ is a Gaussian random variable with unit variance for each time $t$. This would, however, render the observation process discontinuous almost everywhere. It makes sense to require the observation process to be continuous as well, and a simple way to achieve is to take a process $Y(t)$ evolving according to the SDE

SDE: Stochastic Differential Equation

$$dY(t) = CX(t)dt + D^{1/2}dV(t), \tag{2.5}$$

where $V(t)$ is a second Wiener process independent of $W(t)$. Note that, unlike its discrete time counterpart, here $Y(t+dt)$ does not only depend on $X(t)$ but also on $Y(t)$. That does not make the analysis much more complicated though, as one can write the inference in terms of $dY(t)$ just as well.

One can proceed as in the case of discrete time with a small time increment $dt$ and then pass to the limit of $dt \to 0$. This will lead to

$$\mu_{t+dt|t} = (I + Adt)\mu_{t|t},$$

and

$$\Sigma_{t+dt|t} = \Sigma_{t|t} + \left(A\Sigma_{t|t} + \Sigma_{t|t}A^\top + H\right)dt.$$

The distribution of $Y(t+dt)$ in turn is given by

$$P(Y(t+dt)|Y(t),X(t)) = \mathcal{N}\left(Y(t) + CX(t)\,dt, D\,dt\right),$$

or more simply one can directly write down the distribution of $dY(t)$,

$$P(dY(t)|X(t)) = \mathcal{N}\left(CX(t)\,dt, D\,dt\right).$$

The Bayes' update will be

$$P(X(t+dt)|dY(t+dt)) = \frac{P(dY(t+dt)|X(t+dt))P(X(t+dt)|X(t))}{P(dY(t+dt))}.$$

The product of Gaussians will lead to a Gaussian with variance

$$\Sigma_{t+dt|t+dt} = \left(\Sigma^{-1}_{t+dt|t} + C^\top D^{-1} C dt\right)^{-1},$$

which can be Taylor expanded to

$$\Sigma_{t+dt|t+dt} = \Sigma_{t+dt|t} - dt\Sigma_{t+dt|t}C^\top D^{-1} C\Sigma_{t+dt|t} + o(dt^2).$$

Inserting the expression for $\Sigma_{t+dt|t}$ and taking the limit for $dt \to 0$ one obtains the filter equations for $\mu(t) \equiv \mu_{t|t}$ and $\Sigma(t) \equiv \Sigma_{t|t}$. The posterior variance obeys the ordinary differential equation

$$\frac{d\Sigma}{dt} = A\Sigma(t) + \Sigma(t)A^\top + H - \Sigma(t)C^\top D^{-1} C\Sigma(t). \tag{2.6a}$$

The posterior mean, however, is still a stochastic variable, as it is dependent on the diffusion process $Y(t)$. $\mu(t)$ obeys the SDE

$$d\mu(t) = A\mu(t) + \Sigma(t)C^\top D^{-1}\left(dY(t) - C\mu(t)dt\right). \tag{2.6b}$$

The structure of the equations is very similar to the Kalman updates for discrete-time, as the dynamics of the mean only incorporates the observations through an innovation process.

*Kushner-Stratonovich Equation*

In the cases above I could restrict myself to study the mean and covariance because of the linear structure of both the system dynamics and the observation dynamics. In the general case, however, one can not restrict herself to these moments. In the worst-case scenario one can not escape from estimating the full posterior distribution $P(x,t) = P_{X(t)}(x|Y_{0:t}, X(0) = x_0)$ at every time step.

$Y_{0:t} \equiv \{Y(s), 0 \leq s \leq t\}$

For a Markov system with infinitesimal generator $\mathscr{A}$, the unobserved probability density obeys

$$\frac{\partial P(x,t)}{\partial t} = \mathscr{A}^\dagger P(x,t), \tag{2.7}$$

where $\mathscr{A}^\dagger$ is the adjoint of $\mathscr{A}$. If the observation process $Y(t)$ evolves according to

$$dY(t) = c(X(t))dt + D^{1/2}dV(t),$$

then, defining $\hat{c}_t = \int dx\, c(x)P(x,t)$, the posterior distribution obeys the stochastic partial differential equation[13]

$$d_t P(x,t) = \mathscr{A}^\dagger P(x,t)dt + (c(x) - \hat{c}_t)^\top D^{-1}(dY(t) - \hat{c}_t dt)P(x,t). \tag{2.8}$$

Equation (2.8) is usually called the Kushner equation or the Kushner-Stratonovich equation in honor of Harold J. Kushner and Ruslan Stratonovich, the statisticians who first derived it. It is not hard to demonstrate that, by taking $c(X(t)) = CX(t)$ we can recover equation (2.6) above for the evolution of the moments according to equation (2.8).[14] These equations, as one can imagine, are very hard to solve exactly, and approximate solutions are usually employed. The techniques collectively called particle filters seek to generate sample paths $Z(t)$ where the distribution of $Z(t)$ is given by the solution of the Kushner equation. Through a sequential sampling and reweighing procedure this can be done without solving equation (2.8) explicitly.

[13] Here I am using a definition analogous to the definition of $dX(t)$ for a partial difference with respect to time. We have

$$d_t P(x,t) = \lim_{dt \to 0} [P(x, t+dt) - P(x,t)].$$

[14] Bucy, R. S. (1965). Nonlinear filtering theory. *Automatic Control, IEEE Transactions*, 10(2):198

## *2.3 Filtering of Poisson Process Observations*

The theory of filtering of diffusion processes can be extended to the case of Poisson processes as well. Donald Snyder has derived an equation for the filtering of stochastic processes observed through doubly stochastic Poisson processes which bears a remarkable resemblance to equation (2.8).[15] A Pois-

[15] Snyder, D. L. (1972). Filtering and detection for doubly stochastic Poisson processes. *IEEE Transactions on Information Theory*, 18(1):91–102

son process can be defined as a counting process $N(t)$ such that the transition probabilities for infinitesimal times $dt$ are given by

$$P(N(t+dt)-N(t)=0)=1-\lambda dt+o(dt^2) \tag{2.9a}$$

$$P(N(t+dt)-N(t)=1)=\lambda dt+o(dt^2) \tag{2.9b}$$

$$P(N(t+dt)-N(t)>1)=o(dt^2) \tag{2.9c}$$

$$P(N(t+dt)-N(t)<0)=0 \tag{2.9d}$$

In the limit of $dt \to 0$, the transition probabilities for $N_t$ are completely determined by $\lambda$, the rate of the process. Figure 2.1 shows examples of samples from a Poisson process $N_t$ with different rates.

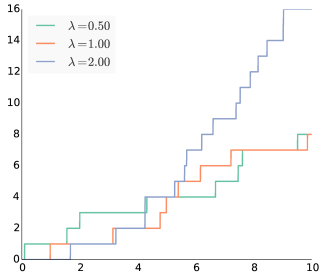


Figure 2.1: Samples of Poisson processes with rates equal to 0.5, 1.0 and 2.0.

A doubly stochastic Poisson process, is a process where the rate $\lambda$ is itself a stochastic random variable, usually a function of another stochastic process $X(t)$. In the case first considered by Snyder, the observations were particle counts of radioactive decay for medical diagnostics. The rate was a function of the concentration of the radioactive substance administered to the patient, and by observing particle counts through time onewould like to infer the concentration of radioactive substance in the patient's organs. The temporal aspect was relevant because of the fast decay of the radioactive particles. In that case one will have

$$P(N(t+dt)-N(t)=0|X(t))=1-\lambda(X(t))dt+o(dt^2) \tag{2.10a}$$

$$P(N(t+dt)-N(t)=1|X(t))=\lambda(X(t))dt+o(dt^2). \tag{2.10b}$$

The conditional probability of a path $N_{0:t}$ given a path $X_{0:t}$ is then

$$P(N_{0:t}|X_{0:t})=\exp\left[-\int_0^t \lambda(X(s))ds+\int_0^t \log(\lambda(X(s)))dN(s)\right], \tag{2.11}$$

where I have used the definition of the stochastic integral with respect to a jump process given in section 2.1. Defining the jump points $t_i$ as the points where $\lim_{t\downarrow t_i} N(t) \neq \lim_{t\uparrow t_i} N(t)$, one can write the usual formula for the Poisson density of spike times

$$P(\{t_i\}|X_{0:t})=\exp\left[-\int_0^t \lambda(X(s))ds\right]\prod_i \lambda(X(t_i)).$$

Armed with a rate model for the DSPP, one can infer the stim-

DSPP: Doubly Stochastic Poisson Process

ulus history from the count history. The posterior distribution for $X_{0:t}$ is

$$P(X_{0:t}|N_{0:t}) \propto P(X_{0:t}) \exp\left[-\int_0^t \lambda(X(s))ds + \int_0^t \log(\lambda(X(s)))dN(s)\right]. \tag{2.12}$$

$P(X_{0:t})$ is a prior distribution over paths of $X(t)$, which in turn are infinite-dimensional objects. This determines the temporal structure of the stimulus, and can be tuned to reflect the statistical properties of the stimuli being considered. In practice, it is very hard to compute the full posterior, and for inference purposes, one generally deals with a discretised version of the path.

Discretising $X_{0:t}$ one can treat the problem of inferring the paths as a multidimensional estimation problem. The simplest way to estimate $X_{0:t}$ is maximum likelihood whereby one maximises the likelihood given in equation (2.11). The function given by equation (2.11) is called a likelihood for $X_{0:t}$ since it does not define a probability density for it. Further, one can incorporate prior beliefs about the structure of $X(s)$ by using the full posterior given in equation (2.12). Taking the value of $X_{0:t}$ that maximises the posterior probability yields the so-called *Maximum a Posteriori* estimator. The full Bayesian approach would be to take the posterior mean as an estimate for $X_{0:t}$, that is one takes the mean of the distribution given in equation (2.12) as our estimator. This is usually very hard to compute and has to be done through sampling methods.[16]

ML = maximum likelihood

MAP = maximum a posteriori

[16] For an extensive review of this so-called *decoding* problem see (Ahmadian et al., 2011; Pillow et al., 2011).

What if one want to estimate the value of $X(t)$ given $N_{0:t}$ in an online fashion? This corresponds to estimating the marginal probability of $X(t)$ according to equation (2.12). We can derive the results of Snyder informally as follows. Using the same notation as in the Kalman case one has

$$P(x,t+dt) = \frac{P\left(X(t+dt)|N_{0:t}\right) P\left(N(t+dt)|X(t+dt)\right)}{P(N(t+dt))}.$$

When unobserved the distribution of $X$ evolves according to equation (2.7). So for infinitesimal $dt$ one can write

$$P(X(t+dt)|N_{0:t}) = P(x,t) + \mathscr{A}^{\dagger} P(x,t)dt.$$

Furthermore, one can write the quotient of terms dependent on $N(t+dt)$ out as a function of $dN(t)$, leading to

$$\frac{P(N(t+dt)|X(t+dt),N_{0:t})}{P(N(t+dt))} = (1-dN(t))\frac{1-\lambda(X(t))dt}{1-\hat{\lambda}dt} + dN(t)\frac{\lambda(X(t))}{\hat{\lambda}},$$

where $\hat{\lambda} = \int dx P(x,t)\lambda(x)$. Expanding and discarding terms of order $dt^2$ and $dN(t)dt$, yields

$$\frac{P(N(t+dt)|X(t+dt),N_{0:t})}{P(N(t+dt))} = 1-dN(t)-\lambda(X(t))dt+\hat{\lambda}dt+dN(t)\frac{\lambda(X(t))}{\hat{\lambda}},$$

which rearranging terms, can be written as

$$\frac{P(N(t+dt)|X(t+dt),N_{0:t})}{P(N(t+dt))} = 1+\left(\lambda(X(t))-\hat{\lambda}\right)\hat{\lambda}^{-1}\left(dN(t)-\hat{\lambda}dt\right).$$

Inserting this into the relation above gives

$$P(x,t+dt) = P(x,t)+\mathscr{A}^{\dagger}P(x,t)dt+P(x,t)\left(\lambda(x)-\hat{\lambda}\right)\hat{\lambda}^{-1}\left(dN(t)-\hat{\lambda}dt\right),$$

or, writing it as a stochastic PDE,

$$d_tP(x,t) = \mathscr{A}^{\dagger}P(x,t)dt+P(x,t)\left(\lambda(x)-\hat{\lambda}\right)\hat{\lambda}^{-1}\left(dN(t)-\hat{\lambda}dt\right). \tag{2.13}$$

Note the striking similarity with equation (2.8), namely the observations only influence the posterior through an innovation process, here given by $dN(t)-\hat{\lambda}dt$. Furthermore, the inverse rate is equivalent to the inverse variance, since the variance of a Poisson process is precisely its rate $\lambda(x)$. This equation was first derived by Donald Snyder in 1972.[17]

Clearly the derivation above is not mathematically sound. More care is needed when taking limits with $dt \to 0$, namely I have ignored the terms of order $dN(t)dt$ in equation (2.13). This can be shown to be rigorous, but is beyond the scope of this thesis.[18] The full derivation by Snyder first finds an expression for the characteristic function of the posterior distribution and then derives a stochastic PDE for the characteristic function. This is then Fourier-transformed to yield equation (2.13).

[17] Snyder, D. L. (1972). Filtering and detection for doubly stochastic Poisson processes. *IEEE Transactions on Information Theory*, 18(1):91–102

[18] See (Privault, 2014) for a full introduction to the stochastic calculus of jump processes.

*Multiple Spike Trains*

Equation (2.13) is readily extended to multiple point processes, as long as they are independent. Given a population of point processes $N^i(t)$, $i \in [1,M]$, one will simply have, following the same derivation

$$d_tP(x,t) = \mathscr{A}^{\dagger}P(x,t)dt+P(x,t)\sum_i\left(\lambda_i(x)-\hat{\lambda}_i\right)\hat{\lambda}_i^{-1}\left(dN^i(t)-\hat{\lambda}_idt\right). \tag{2.14}$$

Again, this can be be compared with a multidimensional observation process in the Kalman case by noting that one can rewrite it as

$$d_tP(x,t) = \mathscr{A}^{\dagger}P(x,t)dt+P(x,t)\left(\boldsymbol{\lambda}(x)-\hat{\boldsymbol{\lambda}}\right)^{\top}Diag(\hat{\boldsymbol{\lambda}})^{-1}\left(d\boldsymbol{N}(t)-\hat{\boldsymbol{\lambda}}dt\right).$$

I have used the notation $\boldsymbol{\lambda}(x) = (\lambda_1(x),\lambda_2(x),\ldots)^{\top}, d\boldsymbol{N}(t) = (dN^1(t),dN^2(t),\ldots)^{\top}$ and so forth. Note that, taking the vector counting process $\boldsymbol{N}(t)$, its covariance will be $Diag(\hat{\boldsymbol{\lambda}})$, and the equation corresponds precisely to equation (2.8).

## 2.4 Fast Population Coding and Dense Tuning Functions

A similar filtering framework was proposed in the computational neuroscience community as well.[19] The main question being asked was how one can extend the framework of population coding, which usually relied on cumulative rates, to coding in a short-time regime. Filtering from spike trains has also been of central importance to the study of Brain-Computer-Interfaces, where one tries to decode intended movements or actions from the activity of neurons in the brain.[20] One issue that is central in the approach of Huys et al. (2007) is the assumption that the population firing rate is independent of the stimulus. I will first extend equation (2.11) to multiple independent Poisson processes. This yields

[19] Huys, Q. J. M., Zemel, R. S., Natarajan, R., and Dayan, P. (2007). Fast population coding. *Neural Computation*, 19(2):404–441

[20] Ergun, A., Barbieri, R., Eden, U. T., Wilson, M. A., and Brown, E. N. (2007). Construction of point process adaptive filter algorithms for neural systems using sequential Monte Carlo methods. *IEEE Transactions on Biomedical Engineering*, 54(3):419–428

$$P(\{N^i_{0:t}\}|X_{0:t}) = \exp\left[\sum_i \int \log(\lambda_i(X(s)))dN^i(s) - \int \lambda_i(X(s))ds\right]. \tag{2.15}$$

If the tuning functions are distributed such that $\sum_i \lambda_i(X) = C$, irregardless of $X$, this can be simplified substantially. This is the same as saying that the process $N(t) = \sum_i N^i(t)$ is a homogeneous Poisson process with rate $C$. One will then have

$$P(\{N^i_{0:t}\}|X_{0:t}) \propto \prod_i \exp\left[\sum_i \int \log(\lambda_i(X(s)))dN^i(s)\right]. \tag{2.16}$$

The integral with respect to $dN^i(s)$ will only yield non-zero terms where $N^i(t)$ is discontinuous, therefore the resulting term will be simply a product of the rates of the neurons at the times they spiked. Let us denote the set of spikes emitted by the population by $\boldsymbol{S}(t) = \{(n_i, t_i)\}_{i=1}^{N(t)}$, where $n_i$ denotes the identitiy of the $i$-th neuron to spike and $t_i$ denotes its spike time. One can then write

$$P(\{N^i_{0:t}\}|X_{0:t}) \propto \prod_{\boldsymbol{S}(t)} \lambda_{n_s}(X(t_s)). \tag{2.17}$$

Furthermore, assume that the the tuning functions $\lambda_i$ are unnormalized Gaussians of the form

$$\lambda_i(x) = \phi \exp\left[-\frac{1}{2}(x-\theta_i)^\top E^\dagger (x-\theta_i)\right], \tag{2.18}$$

where I have used the pseudoinverse $E^\dagger$ to allow for the tuning functions to be degenerate Gaussian distributions. This poses no problem, as the prior over $X(t)$ will be chosen to be Gaussian, leading to a Gaussian posterior when multiplied by $\lambda_i / \hat{\lambda}_i$. Furthermore the marginal rate of a spike being fired $\hat{\lambda}_i = \boldsymbol{E}_X[\lambda_i(x)]$ is also defined. One must note that $\lambda_i$ does

not define a distribution over the stimulus space but a rate of arrival of observations. The Gaussian updates are, however the same.

I can now treat the problem similarly to the Kalman filter problem, but one needs to take into account the fact that instead of arriving continuously, observations are coming in at random times. Consider the same process as before given by the SDE

$$dX(t) = AX(t)dt + H^{1/2}dW(t).$$

In the absence of observations the Gaussian distribution will evolve as

$$\frac{d\mu}{dt} = A\mu \tag{2.19a}$$

and

$$\frac{d\Sigma}{dt} = A\Sigma + \Sigma A^{\top} + H. \tag{2.19b}$$

Therefore, the posterior distribution over $X(t)$ between observations is given by $\mathcal{N}(\mu(t), \Sigma(t))$. If a neuron with tuning centre $\theta_i$ spikes at time $t$, the posterior density will be updated by

$$P(X(t)|\text{spike}) = \frac{\lambda(X(t))\mathcal{N}(\mu(t), \Sigma(t))}{\hat{\lambda}}.$$

Completing squares in the exponents, one obtains for the posterior mean

$$\begin{aligned}\mu(t) = &\quad \left(\Sigma(t^-)^{-1} + E^{\dagger}\right)^{-1}\left(\Sigma(t^-)^{-1}\mu(t^-) + E^{\dagger}\theta_i\right)\\ = &\quad \mu(t^-) - \left(\Sigma(t^-)^{-1} + E^{\dagger}\right)^{-1}\left(\Sigma(t^-)^{-1} + E^{\dagger}\right)\mu(t^-)\\ + &\quad \left(\Sigma(t^-)^{-1} + E^{\dagger}\right)^{-1}\left(\Sigma(t^-)^{-1}\mu(t^-) + E^{\dagger}\theta_i\right),\end{aligned}$$

and finally

$$\mu(t) = \mu(t^-) + \Sigma(t)^{-1}E^{\dagger}\left(\theta_i - \mu(t^-)\right). \tag{2.20a}$$

For the covariance one has

$$\begin{aligned}\Sigma(t) = &\quad \left(\Sigma(t^-) + E^{\dagger}\right)^{-1}\\ = &\quad \Sigma(t^-) - \Sigma(t^-)\left(\Sigma(t^-) + E^{\dagger}\right)\left(\Sigma(t^-) + E^{\dagger}\right)^{-1} + \left(\Sigma(t^-) + E^{\dagger}\right)^{-1},\end{aligned}$$

yielding

$$\Sigma(t) = \Sigma(t^-) - \Sigma(t^-)E^{\dagger}\Sigma(t^-)\left(I + E^{\dagger}\Sigma(t^-)\right)^{-1}. \tag{2.20b}$$

These updates can be simplified if the tuning matrix $E$ is invertible.

These equations can be condensed into SDE's for the posterior mean and covariance very simply. The mean will be given by

$$d\mu(t) = A\mu(t)dt + \sum_i dN^i(t)\left[\Sigma(t^-)\left(I + E^{\dagger}\Sigma(t^-)\right)^{-1}E^{\dagger}\left(\theta_i - \mu(t^-)\right)\right] \tag{2.21a}$$

and the covariance by

$$d\Sigma(t) = (A\Sigma(t) + \Sigma(t)A^\top + H)dt + dN(t)\left[\Sigma(t^-)E^\dagger\Sigma(t^-)\left(I + E^\dagger\Sigma(t^-)\right)^{-1}\right]. \tag{2.21b}$$

These SDE's define processes that are continuous from the right and have a limit from the left. They are often called Càdlàg processes in the stochastic literature, from the french phrase *continue à droite, limite à gauche*. The evolution of the posterior variance only depends on the total spike count process $N(t)$, which will be fundamental for the future analysis.

As I mentioned before, the covariance of the tuning functions does not need to be invertible. Note that as long as $\Sigma(t)^{-1} + E^\dagger$ is invertible, the filtering equations are always well-defined. This can be ensured by requiring that $E$ be positive semidefinite. Since $\Sigma(t)$ is positive definite, as it is a covariance matrix, $\Sigma(t)^{-1} + E^\dagger$ will also be positive definite.

Most of the analytic work in this thesis is done on the filtering problem given by equation (2.21). The fact that the total frequency of observations is independent of the system's state along with the homogeneous nature of the population of processes leads to a number of simplifications when evaluating the Mean-Squared-Error of the estimator $\mu(t)$. More specifically, since $\mu(t)$ is the posteriori mean estimator, its MSE is given by the average posterior variance.[21] The filtering scheme described in this section and some of the results of chapter 3 are illustrated in figure 2.2.

Mean-Squared-Error ≡ MSE

[21] This will be shown in the beginning of chapter 3.

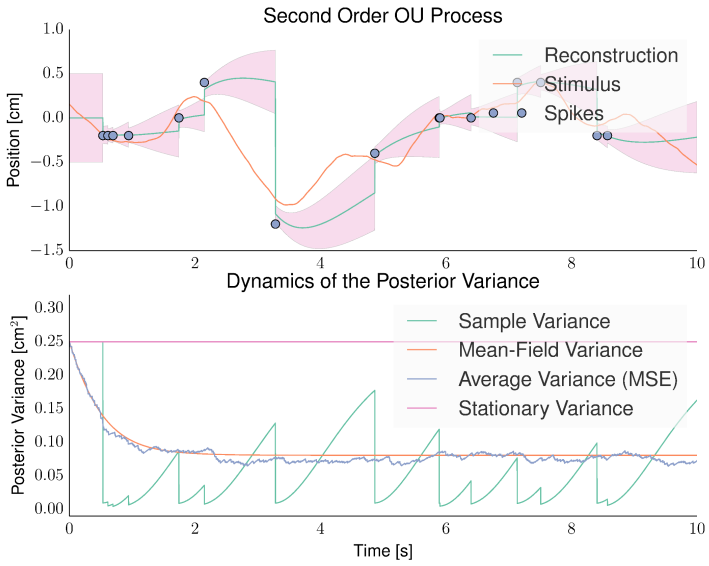


Figure 2.2: The general filtering framework: the unobserved process we are trying to estimate is shown as the solid red line, while the observed spikes are shown as red dots, aligned by the preferred stimulus of the firing neurons. The posterior mean estimate is given by the dotted blue line, while the light red shading gives the confidence interval of one standard deviation. Note the discontinuous jumps in the mean and covariance at the times of spikes. The lower figure shows the average posterior variance over all possible spike trains given a stimulus distribution. Note that the mean-field approximation provides a very good account of the evolution of the average. These results will be further discussed in chapter 3.

I will now turn to filtering from Point processes when the dense coding assumption does not hold.

### 2.5 Methods for General Filtering of Point Processes

If nothing else is known about the process at hand, one is forced to work directly with equation (2.14). In principle one could discretise the state space and try to solve the Partial Differential Equation recursively as the observations come in. In practice, however, the right hand side equation (2.14) also contains averages over $P(x,t)$, leading to additional complications on every integration step. One way to circumvent this particular problem is to work with unnormalized probabilities. The Zakai equation[22] is a modified version of the Kushner equation which propagates unnormalized probabilities. For a stochastic process $X(t)$ observed through another process $Y(t)$ as given in section 2.2, one can define $\rho(x,t)$ as a solution to

Partial Differential Equation ≡ PDE

[22] Zakai, M. (1969). On the optimal filtering of diffusion processes. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 11(3):230–243

$$d_t\rho(x,t) = \mathscr{A}^\dagger\rho(x,t)dt + \rho(x,t)x^\top C^\top D dY(t), \quad (2.22)$$

with $\rho(x,0) = P_0(x)$. It can then be shown that $P(x,t) = \rho(x,t)/\int dx\rho(x,t)$. Any solution to the Zakai equation will yield a solution to the corresponding Kushner equation when normalised. Note that, while the Kushner equation was a stochastic partial integro-differential equation, since the left hand side involved averages over $P(x,t)$, the Zakai equation is a simpler linear stochastic partial differential equation and given a realisation of the observation process can be solved by standard PDE methods.

I will present a similar framework for the Snyder equation 2.13. Again taking the notation $P(x,t) = \rho(x,t)/\int dx\rho(x,t)$ one finds that the unnormalised posterior distribution $\rho(x,t)$ of a stochastic process with generator $\mathscr{A}$ observed through a doubly stochastic Poisson process with rate $\lambda(x)$ will obey the stochastic PDE

$$d_t\rho(x,t) = \mathscr{A}^\dagger\rho(x,t)dt - \lambda(x)\rho(x,t) + (\lambda(x)-1)\,\rho(x,t)dN(t). \quad (2.23)$$

Note that any term independent of $x$ can be trivially discarded as it only constitutes a temporal renormalisation of $\rho$. For example, if $\rho^*(x,t)$ is a solution to equation (2.23) with initial condition $\rho(x,0) = g(x)$, then $r(x,t) = \exp\left(-\int_0^t k(s)ds\right)\rho^*(x,t)$ is a solution to the stochastic PDE

$$d_t r(x,t) = \mathscr{A}^\dagger r(x,t)dt - \lambda(x)r(x,t) + (\lambda(x)-1)\,r(x,t)dN(t) - k(t)r(x,t)dt,$$

with the same initial condition. This allows one to set a baseline to the expected firing rate in the unnormalised equation. This framework has been used by (Bobrowski et al., 2009) in the study of finite state systems observed through doubly stochastic Poisson processes. This work was also extended to static continuous processes by Yaeli and Meir.[23]. I will now discuss the application of these equations to the development of a particle filter for the filtering problems discussed above.

[23] Yaeli, S. and Meir, R. (2010). Error-based analysis of optimal tuning functions explains phenomena observed in sensory neurons. *Frontiers in computational neuroscience*, 4(October):16

*Particle Filtering*

The central idea of particle filtering is relatively simple. If one is not given access to the system's state directly, one can just simulate a large number of hypotheses of the system's state and weight each copy according to its agreement with the observations. One can then compute averages over the posterior distribution from the weighted samples. Say I have a system with state $X(0)$ initially distributed according to $P_0(x)$, some known transition probability

$$W(x,x') \equiv P_{X(t)}\left(x\middle|X(t)=x'\right),$$

and I am given observations of a second process $Y(t)$, with probability density

$$\mathscr{L}(y,x) \equiv P_{Y(t)}\left(y\middle|X(t)=x\right).$$

If all the probabilities are known I can implement the filtering steps numerically, by taking a sample of $M$ *particles* $\{Z^i(0)\}, i \in [1,\ldots,M]$ from $P_0(x)$, and associating a weight to each of those particles $w^i(0) = 1$. Then for each particle $Z^i$ I sample the state of that particle at the following instant through the transition probability

$$P_{Z^i(t)}(z|Z^i(t-dt)) = W(z, Z^i(t-dt))$$

and reweigh it through the likelihood of $Y(t)$, yielding

$$w^i(t) = w^i(t-dt)\mathscr{L}(Y(t), Z^i(t)).$$

The approximate density $Q(x,t) = \sum_i w^i(t)\delta(Z^i(t)-x)/\sum_i w^i(t)$, then gives an approximation of the posterior density $P(x,t)$ and averages can be computed by simply aggregating over the particles, giving

$$\int dx\, P(x,t)g(x) \approx \int dx\, Q(x,t)g(x) = \frac{1}{\sum_i w^i(t)}\sum_i w^i(t)g(Z^i(t)).$$

These methods are often called Sequential Monte Carlo methods, since they consist of sequentially sampling the state of the system in a way similar to a Monte Carlo Markov Chain.

SMC: Sequential Monte Carlo

The description above barely scratched the surface of what is achievable and what are the problems of particle filters, and I will not dive too deeply into the theory of them, but one point should be made. Though the sampling procedure described above in principle yields an estimate of the true posterior distribution, a lot can go wrong when implementing it with a finite number of particles. One issue that plagues many such filters is the issue of weight depletion. Weight depletion refers to the situation where all but a few particles have very low weights, representing state paths which are incompatible with the observations. This can lead the particle filter to

waste resources estimating the density of regions which don't contribute to the posterior averages, and therefore yielding very poor estimates of the distribution in the interesting regions. This led researchers to propose resampling steps in the particle filter. Whenever a certain criterion is met (or after every step in the filter) one can resample the particles from the set of existing particles according to their weights, i.e., sample $M$ particles from the set $\{Z^i(t)\}$ with probabilities given by $p_i = w^i(t)/\sum_i w^i(t)$. After that, all weights are reset to 1 and the procedure continues. This forces the filter to allocate its particles according to its current estimate of the posterior distribution, preventing weight depletion to some extent. It is not a panacea for these issues, however, and even properly resampled filters can often end up with very poor estimates of the posterior distribution.

Another important thing to note, is that it is often not possible to efficiently sample from the transition probabilities of the system. In those cases one can still combine the particle filter with an importance sampling approach. In that sense, at every step one samples from a simpler distribution $Q(Z^i(t)|Z^i(t-dt))$ and reweighs the particles according to

$$w^i(t) = w^i(t-dt)\frac{P(Y(t)|Z^i(t))P(Z^i(t)|Z^i(t-dt))}{Q(Z^i(t)|Z^i(t-dt))}.$$

This allows for efficient sampling, but it adds another source of weight depletion. Again, if the sampling transition probabilities do not match the system's transition probabilities, the weights will quickly fall to low values, leading to poor estimates of the posterior distribution.

Let us consider again the general case of doubly stochastic Point process filtering. Note that in the absence of spikes the posterior evolves according to

$$\frac{\partial P(x,t)}{\partial t} = \mathscr{A}^\dagger P(x,t) + (\hat{\lambda}(t) - \lambda(x,t))P(x,t).$$

For linear diffusion processes, this simplifies to

$$\frac{\partial P(x,t)}{\partial t} = -\nabla\cdot(AxP(x,t)) + \frac{1}{2}\mathrm{Tr}\left[H\frac{\partial^2 P(x,t)}{\partial x_i \partial x_j}\right] + (\hat{\lambda}(t) - \lambda(x,t))P(x,t). \tag{2.24}$$

I can again define an unnormalised density $\rho(x,t)$ evolving according to

$$\frac{\partial \rho(x,t)}{\partial t} = -\nabla\cdot(AxP(x,t)) + \frac{1}{2}\mathrm{Tr}\left[H\frac{\partial^2 P(x,t)}{\partial x_i \partial x_j}\right] - \lambda(x,t)\rho(x,t),$$

for which the normalised density $\rho(x,t)/\int dx\rho(x,t)$ satisfies equation (2.24). It can be shown that the equation

above describes the evolution of a drift diffusion process with a death rate of $\lambda(x,t)$. This means that the system evolves according to the equation (2.4) but there is a transition to a death state with a rate $\lambda(x,t)$.[24] This allows one to formulate a simple particle filter, by propagating the particles with the transition probability of the linear stochastic system and then killing it at a rate $\lambda(Z^i(t),t)$, resampling the particles every time a particle *dies*. Alternatively one can reweigh the weights according to $1-\lambda(Z^i(t),t)dt$ after every time step, obtaining the same effect.

The particle filtering scheme presented here is very flexible, and is in principle applicable to any kind of stochastic process observed through Poisson spikes. This approach has also gained traction in the neuroscience community, where particle filters are often used to decode cortical signals from electrophysiological recordings.[25] Most BCI application require very low latency though, and often specialised types of Kalman filter are more practical to employ in such settings.[26]

[24] Ø ksendal, B. (2003). *Stochastic Differential Equations: An Introduction with Applications*, volume 10 of *Universitext*. Springer

[25] Brockwell, A., Rojas, A., and Kass, R. (2004). Recursive bayesian decoding of motor cortical signals by particle filtering. *Journal of Neurophysiology*, 91(4):1899–1907; and Ergun, A., Barbieri, R., Eden, U. T., Wilson, M. A., and Brown, E. N. (2007). Construction of point process adaptive filter algorithms for neural systems using sequential Monte Carlo methods. *IEEE Transactions on Biomedical Engineering*, 54(3):419–428

[26] Wu, W., Gao, Y., Bienenstock, E., Donoghue, J. P., and Black, M. J. (2006). Bayesian population decoding of motor cortical activity using a kalman filter. *Neural computation*, 18(1):80–118

*Assumed Density Filtering*

Though the presented framework of DSPP's in dense Gauss-Poisson populations of neurons turns out to be exactly Gaussian, this does not hold generally. For example, one could have a stimulus-dependent population firing rate, leading to non-Gaussian posteriors. One would then have to deal with the full extent of the Snyder equation (2.14). One way to deal with this is to project the posterior distribution to a Gaussian at every time step, that is, at every time $t$, one looks at the resulting distribution at the next time step $t+dt$ and approximates it with a Gaussian. To do so one needs to determine the mean and covariance of the posterior and can then match a Gaussian distribution to those moments.

This approach is usually called Assumed Density Filtering. Given some variable of interest $x$ and a set of observations of random variables $\{Y_1,\ldots,Y_N\}$ distributed as $P_Y(y|x)$, ADF consists of sequentially incorporating the observations and finding the best approximation to the posterior within a family of distributions. For example, if the true, intractable distribution were $P(x|Y_1,\ldots,Y_N)$ one could choose to approximate it by a Gaussian distribution. One would start out with a prior distribution $Q_0(x)$ and sequentially look for the best Gaussian approximation to the posterior $Q_i(x)P(Y_{i+1}|x)$. This is usually termed filtering even when there is no temporal estimation involved because of the sequential updates to the posterior.[27] The best approximation to the posterior is usually defined as the one minimising the Kullback-Leibler divergence between the full and approximate posterior. In that sense, given a current approximation $Q_i(x)$ and a new obser-

ADF: Assumed Density Filtering

[27] For examples of applications, see (Opper, 1998; Boyen and Koller, 1998; Minka, 2001).

vation $Y_{i+1}$, the update to our approximate posterior would be

$$\begin{aligned} Q_{i+1}(x) &= \text{argmin}_q \, KL[Q_i(x)P(Y_{i+1}|x)||q] \\ &= \text{argmin}_q \int dx \, Q_i(x)P(Y_{i+1}|x) \log \frac{Q_i(x)P(Y_{i+1}|x)}{q(x)}. \end{aligned}$$

The KL divergence taken here is the reverse of the KL divergence used in variational inference.[28] It can be shown that if one applies this process using a family of exponential distributions as approximating distributions, it will lead to a moment matching procedure where the moments of the approximating distribution match the ones of the posterior $Q_i(x)P(Y_{i+1}|x)$.[29] If one took $Q(x)$ to be a Gaussian in every step, the procedure would involve evaluating the mean and covariance of the posterior and setting the new distribution to a Gaussian with that mean and covariance.

[28] In variational inference one usually considers the KL divergence between the approximating and the true distribution given by $KL[q||p] = \int dx q(x) \log \frac{q(x)}{p(x)}$. When $q(x)$ is tractable or allows for exact integration, this allows for simplifications of the KL-divergence.

[29] See appendix A.2 for a short clarification.

A simple example of a factor leading to a non-Gaussian posterior in the filtering problem described in this chapter is the presence of adaptation in the firing rates. Poisson processes are memoryless, that is, the probability of a spike being fired is independent of the time since the last spike. It is well known, however, that biological neurons do not follow that rule. For example, there is a clear refractory period in action potential generation, rendering a neuron incapable of firing an action potential for a short period after the firing of an action potential, regardless of the stimulation applied. This refractory period varies from cell type to cell type and between organisms, but is generally around 5 ms. Another very common phenomenon is spike-frequency-adaptation,[30] where upon continued stimulation a neuron reduces its frequency from its initial response frequency to a lower frequency. A simple Poisson process can not account for these phenomena, but it is easy to modify the Poisson model to account for a refractory period or else to include a spike-frequency adaptation component as well.

[30] Benda, J. and Herz, A. V. (2003). A universal model for spike-frequency adaptation. *Neural computation*, 15(11):2523–2564

Consider a simple history-dependent Poisson process given by a rate $\lambda(x,t) = \kappa(t)\lambda(x)$, where $\kappa$ itself depends on the spiking history of the process. Let me take $\kappa$ evolving according to the SDE

$$d\kappa(t) = \frac{(\phi - \kappa(t))}{\tau} dt - h(\kappa(t))dN(t), \quad h(\kappa) = \min(\Delta, \kappa),$$

where $dN(t)$ is the spike train of the neuron. This will lead to a rate modulation which stabilises at $\phi$ when there are no spikes, and is shifted downwards by $\Delta$ whenever there is a spike, without venturing below 0. Although the process is now history-dependent, the joint process $\kappa(t), N(t)$ is still Markov, since the dynamics of $\kappa$ itself is Markovian. This allows one to model a neuron with a refractory period by taking

a relaxation time $\tau \approx 5ms$ or to model a neuron with spike-frequency-adaptation by taking longer relaxation times.

The filtering probability for a diffusion process observed through a population of adaptive neurons with rates given by $\lambda^i(x,t) = \kappa^i(t)\lambda^i(x)$ is given by

$$d_t P(x,t) = \mathscr{A}^\dagger P(x,t)dt + P(x,t)\sum_i \left(\lambda_i(x,t) - \hat{\lambda}_i(t)\right)\hat{\lambda}_i(t)^{-1}\left(dN^i(t) - \hat{\lambda}_i(t)dt\right), \tag{2.25}$$

which is equation (2.13) with time-dependant firing rates. One now needs to integrate the set of equations for the rate modulations $\kappa^i(t)$ for every neuron as well, to be able to solve the Snyder equation properly.

The ADF approach for this case would work as follows: one starts out with an initial Gaussian distribution $\mathscr{N}(\mu(0), \Sigma(0))$ at $t = 0$; then, for every instant $t$ one determines the non-Gaussian probability P(x,t+dt) at the next instant $t + dt$ via equation (2.25); after that, one finds the mean and covariance of $P(x, t+dt)$, and approximates the distribution by a Gaussian with the same mean and covariance and proceeds to the next instant. This can be cast into a set of differential equations and updates governing the mean and covariance of our approximate posterior.

To obtain the ADF equations for this simple model I need to evaluate the evolution of the mean and covariance of the filtering distribution. I will derive the necessary equations similarly to the derivation of the differential Chapman-Kolmogorov equation in (Gardiner, 2004).The average of a function of $x$ over the posterior distribution evolves as

$$\frac{\partial \boldsymbol{E}_P[f]}{\partial t} = \frac{\partial \int dx f(x)P(x,t)}{\partial t} = \lim_{dt\to 0}\frac{\int dx f(x)\left(P(x,t+dt) - P(x,t)\right)}{dt}.$$

In the absence of spikes, the limit can be evaluated, as all terms are of order $dt$, obtaining

$$\begin{aligned}\frac{\partial \boldsymbol{E}_P[f]}{\partial t} &= \int dx f(x)\left(\mathscr{A}^\dagger P(x,t) + (\hat{\lambda}(t) - \lambda(x,t))P(x,t)\right) \\ &= \int dx P(x,t)\left(\mathscr{A} f(x) + f(x)(\hat{\lambda}(t) - \lambda(x,t))\right).\end{aligned}$$

This can be readily cast into a form to allow for moment matching of Gaussian distributions. Taking a stochastic process $X(t)$ given by the SDE

$$dX(t) = A(X(t))\,dt + H(X(t))^{1/2}\,dW(t),$$

the infinitesimal generator and its adjoint will be given by

$$\mathscr{A} f = A(x)^\top \nabla f(x) + \frac{1}{2}\mathrm{Tr}\left[H(x)\frac{\partial^2 f(x)}{\partial x^2}\right],$$

and

$$\mathscr{A}^{\dagger} f = -\nabla \cdot (A(x) f(x)) + \frac{1}{2} \operatorname{Tr}\left[ \frac{\partial^2 H(x) f(x)}{\partial x^2} \right].$$

The evolution of the mean and covariance will thus be given by

$$\frac{\partial \mu(t)}{\partial t} = \boldsymbol{E}\left[A(x)\right] + \boldsymbol{E}\left[x\left(\hat{\lambda}(t) - \lambda(x,t)\right)\right], \tag{2.26a}$$

$$\begin{aligned}\frac{\partial \Sigma(t)}{\partial t} =& \boldsymbol{E}\left[A(x)(x-\mu(t))^{\top}\right] + \boldsymbol{E}\left[(x-\mu(t))A(x)^{\top}\right] + H(x) \\ &+ \boldsymbol{E}\left[(x-\mu(t))(x-\mu(t))^{\top}\left(\hat{\lambda}(t) - \lambda(x,t)\right)\right].\end{aligned} \tag{2.26b}$$

These equations are exact, even if the posterior distribution is not Gaussian. If the posterior is Gaussian, the averages on the right hand side of these equations can be written as a function of $\mu(t)$ and $\Sigma(t)$, therefore providing a closed system for the evolution of these variables. The crucial step to perform ADF is to assume that the distribution at every instant is characterised by only its mean and covariance, and is therefore Gaussian. In that case, the averages in the equations can often be performed exactly and one can provide an approximate filter to the problem. Note that the derivation is valid for the case of multiple spike trains as well, yielding

$$\frac{\partial \mu(t)}{\partial t} = \boldsymbol{E}\left[A(x)\right] + \sum_i \boldsymbol{E}\left[x\left(\hat{\lambda}^i(t) - \lambda^i(x,t)\right)\right], \tag{2.27a}$$

$$\begin{aligned}\frac{\partial \Sigma(t)}{\partial t} =& \boldsymbol{E}\left[A(x)(x-\mu(t))^{\top}\right] + \boldsymbol{E}\left[(x-\mu(t))A(x)^{\top}\right] + H(x) \\ &+ \sum_i \boldsymbol{E}\left[(x-\mu(t))(x-\mu(t))^{\top}\left(\hat{\lambda}^i(t) - \lambda^i(x,t)\right)\right].\end{aligned} \tag{2.27b}$$

In chapter 5 I will apply the ADF approach to the general linear stochastic systems considered here as well as a nonlinear stochastic system and compare them to the particle filter approach. Though the ADF has had considerable success and has spawned a number of new approaches, most notably the expectation propagation (EP) algorithm,[31] the theoretical guarantees of particle filters have led me to prefer it when estimating the MSE of an approximate filter.

[31] Opper, M. and Winther, O. (2000). Gaussian processes for classification: Mean-field algorithms. *Neural Computation*, 12(11):2655–2684; and Minka, T. P. (2001). Expectation propagation for approximate bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc

## *2.6 Filtering for General Gaussian Processes*

The linear stochastic processes I have considered in this chapter are special cases of Gaussian Processes. A Gaussian Process is a process $X(t)$ such that the marginal distribution of

the process at a set of times $\{t_1,\ldots,t_M\}$ is always given by a Gaussian distribution. Furthermore, the density of $X(t)$ at said points is given by

$$P\left(X(t_1),X(t_2),\ldots,X(t_m)\right)=\mathcal{N}\left((m(t_1),m(t_2),\ldots,m(t_M))^\top,K(t_i,t_j)\right),$$

where $1\leq i\leq M,\quad 1\leq,j\leq M$. The covariance of a GP is given by the kernel function $K(s,t)$, which specifies the temporal structure of the process at hand. It is straightforward to show that the unobserved Ornstein-Uhlenbeck process

GP ≡ Gaussian Process

$$dX(t)=-\gamma X(t)dt+\sigma^{1/2}dW(t),$$

describes a Gaussian process with zero mean and kernel

$$K_{OU}(s,t)=\frac{\sigma}{2\gamma}e^{-\frac{|t-s|}{2\gamma}}.$$

Gaussian Processes have become a very popular method in Machine Learning, as they allow one to specify a distribution of random functions over a domain.[32]

[32] Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 1st edition

Assume one is trying to estimate a function $f(t)$ drawn from a GP prior with zero mean and kernel $k(s,t)$. If one is given $M$ observations $(t_i,y_i)$ of the value of $f$ at times $t_i$, one can write the marginal distribution of $f(t)$ for any time $t$ by simple manipulation of Gaussian densities. If the observations are corrupted with Gaussian noise with variance $\alpha^2$, the probability density of the observations is given by

$$P(y_1,\ldots,y_M)=\mathcal{N}(\mathbf{0},K(t_i,t_j)+\alpha^2\delta_{i,j}),\text{ where }1\leq i\leq M,\quad 1\leq,j\leq M.$$

The joint density of $f(t)$ and the observations is given by

$$P(f(t),y_1,\ldots,y_M)=\mathcal{N}\left(\mathbf{0},\begin{bmatrix}K(t,t) & K(t,t_i)\\ K(t,t_j) & K(t_i,t_j)+\alpha^2\delta_{i,j}\end{bmatrix}\right).$$

Let $G_{i,j}=K(t_i,t_j)$, $\boldsymbol{y}=(y_1,\ldots,y_M)^\top$ and $k(t,\{t_i\})=(K(t,t_1),\ldots,K(t,t_M))^\top$. The conditional distribution of $f(t)$ given the observations can then written as

$$P(f(t)|y_1,\ldots,y_M)=\mathcal{N}\left(\hat{f}(t),\Xi(t,t)\right),$$

where

$$\hat{f}(t)=k(t,\{t_i\})^\top(G+\alpha^2\boldsymbol{I})^{-1}\boldsymbol{y},\tag{2.28a}$$

and

$$\Xi(t,t)=K(t,t)-k(t,\{t_i\})^\top(G+\alpha^2\boldsymbol{I})^{-1}k(t,\{t_j\})^\top.\tag{2.28b}$$

As I have shown above, if $t>t_i\forall i$, s.t, $1\leq i\leq M$, these relations can be cast into the form of stochastic differential equations for a number of kernels. The OU kernel and its

corresponding SDE were shown above, but another example I will refer to is the Matern kernel of order $\nu = 3/2$, given by

$$K_{Mat}(t,s) = \eta\left(1 + \frac{\sqrt{3}|t-s|}{l}\right)e^{-\frac{\sqrt{3}|t-s|}{l}}.$$

The samples of the kernel correspond to a critically damped stochastic oscillator with a white-noise force being applied to it. Samples from this process can be seen in figure 5.2. I have chosen the scaling factor to obtain the same characteristic length as the RBF kernel below. It can be shown that an appropriate limit of Matern kernels of increasing order will converge to the RBF kernel (see (Rasmussen and Williams, 2005)).

The approach developed in the beginning of this chapter is very practical as it allows us to use the tools of stochastic dynamics to analyse the expected mean-squared error of the optimal filter, but in the general case of GP's this is not possible. In the case of smoother GP's such as the ones given by the RBF kernel

$$K_{rbf} = \eta \exp\left[-\frac{|t-s|^2}{2l^2}\right],$$

the future covariance depends on all past observations, and one can not formulate simple Markov dynamics for the posterior variance. I will develop a theory for the evolution of the entire posterior kernel

$$\Xi(t,s) = \boldsymbol{E}\left[(f(t)-\hat{f}(t))(f(s)-\hat{f}(s))\middle|\boldsymbol{y}\right]$$

in the next chapter, which allows one to evaluate the average performance of the optimal filter on a general GP observed through Poisson spikes. The learning performance of GP regression methods is still an active area of research, and recent efforts using methods from statistical physics of disordered systems have shown promising advances.[33]

[33] See (Malzahn and Opper, 2005; Urry and Sollich, 2013).

# 3

# *Mean-Squared-Error for Point Process Filtering*

> Well, my theory is that your theory is wrong.
>
> Fernanda G. P. Susemihl

In the previous chapter I have introduced filtering methods for stochastic processes observed through Poisson processes. In this chapter I will deal with the issue of how well one can estimate that process from a set of Poisson processes with a given set of rate functions.

If one has an estimate $\hat{X}$ of some stochastic variable $X$ to be estimated, how can one quantify the error incurred by the estimator? One needs to specify some loss function that assigns a cost to an error $|\hat{X}-X|$. It makes sense to require the loss function to be increasing and to require the loss of an exact estimate to be zero, but other than that, any loss function could do. Here I will consider results on the squared loss function

$$L^2(\hat{X},X) = (\hat{X}-X)^2,$$

leading to the mean-squared-error (MSE) as a measure of the performance of the encoder. This is specially convenient when dealing with Gaussian distributions, as a number of analytical results can be derived. Other possible choice would be the absolute loss, given by

$$L^1(\hat{X},X) = |\hat{X}-X|.$$

In other situations, such as classification tasks, other loss functions are useful, but I will restrict myself to the MSE case here.

When dealing with stochastic processes, it is usually more convenient to consider the covariance matrix of the process, rather than the squared error, which is given by the sum of the variances of every coordinate of the process. Let me then define the mean-squared error matrix as

$$\boldsymbol{MSE}(\hat{X}) = \boldsymbol{E}\left[(\hat{X}-X)(\hat{X}-X)^\top\right],$$

where the expectation is over all possible realisations of $X$ and over the observations leading to the estimator $\hat{X}$. Alter-

natively one could take the average to be over multiple realisations of an experiment, or over long time averages of a temporal estimation problem. To obtain a scalar measure of the estimation error, one can just take the trace of the MSE matrix. This scalar error will be written as

$$MSE = \mathrm{Tr}(\boldsymbol{MSE}) = \boldsymbol{E}\left[(\hat{X}-X)^\top(\hat{X}-X)\right].$$

If further one knows that the estimator $\hat{X}$ depends on some parameters $\theta$, she could seek out the optimal estimator $\hat{X}^*$ by taking the parameters $\theta^*$ that minimise the MSE

$$\theta^* = \mathrm{argmin}_\theta MSE(\hat{X}(\theta)).$$

Assuming I am estimating $X$ from an observation process $Y$ dependent on $X$, I can write

$$MSE(\hat{X}(Y)) = \int dX\,dY\,(\hat{X}(Y)-X)^\top(\hat{X}(Y)-X)P(X|Y)P(Y). \tag{3.1}$$

Since $P(Y) \geq 0$ for every $Y$, minimising the inner integrand $\int dX(\hat{X}(Y)-X)^\top(\hat{X}(Y)-X)P(X|Y)$ for every $Y$ will lead to a minimum of the full integral. So, minimising the inner integrand with respect to the estimator will lead to

$$\frac{\partial \int dX(\hat{X}(Y)-X)^\top(\hat{X}(Y)-X)P(X|Y)}{\partial \hat{X}(Y)} = 2\int dX(\hat{X}(Y)-X)P(X|Y).$$

Equating the derivative to zero leads to the minimum mean squared-error (MMSE) estimator for $X$, given by

$$\hat{X}^*(Y) = \int dX\,X\,P(X|Y) = \boldsymbol{E}[X|Y].$$

The MMSE estimator is often called the Bayes estimator for $X$ given the observations $Y$, since it minimises an expected loss given by equation (3.1), which treats the quantity being estimated as a random variable with a prior probability distribution. The MMSE estimator, however, can only be exactly computed if the true data generating distribution $P(Y|X)$ along with the true signal distribution $P(X)$ is known, allowing one to estimate the posterior probability $P(X|Y)$. Furthermore, the Bayes estimator involves averaging over the signal space, which can be impractical. A number of techniques can be used to approximate the estimator though, such as Gaussian processes or neural networks. The optimality of the MMSE estimator is a central result in information theory, and the MMSE estimator is usually taken as the standard to estimation.[1]

However, finding the optimal estimator is not the end of the story. Often the design of sensors and of the experimental

[1] Minimising a different loss function or error measure would lead to a different optimal estimator. Minimising the expected absolute loss, for example, leads to the posterior median as the optimal estimator.

process allow one to change the data generating distribution $P(Y|X)$. For a simple example, consider a radar gun. Assume it gives a measure of the speed of the considered vehicle corrupted with Gaussian noise with zero mean and standard deviation of 5 km/h. Indeed, given a number of measurements of the speed of a vehicle,[2] the MMSE estimator will be the estimator which minimises the expected MSE. Regardless of that, however, one can always reduce the MSE by using a radar gun with a smaller noise rate. A superior radar gun which outputs measurements with standard deviation of 1 km/h will certainly reduce the MSE further. In most simple cases, however, this reduction is obvious, as one simply strives to reduce the noise as much as possible.

[2] Assuming the speed of the vehicle remained unchanged across measurement

The neural case poses an interesting exception though. Considering the Poisson model from the previous chapter, the probability of a spike being fired in a small time interval $dt$ conditioned on the stimulus $X$ is given by $\lambda(X)dt$. The probability of a spike being fired averaged over all stimuli $X$ would then be given by $\hat{\lambda}dt = \int dx\, P(x)\, \lambda(x)dt$. If one tries to increase the precision of the likelihood defined by $\lambda(X)$, for example by reducing the width of the tuning function, this will automatically reduce the probability of that neuron firing. Therefore, there is a trade-off between frequency of firing and precision of firing, which is not present in the case of additive Gaussian noise. This is illustrated in figure 3.1, where I show two Poisson neurons with Gaussian tuning functions and the same preferred stimulus but with different precisions. The upper neuron has a broader tuning function, leading to a higher firing rate, but in turn the spikes are less discriminating of the value of the stimulus. The second neuron has a much narrower tuning function, leading to more precise observations of the stimulus, but a much lower spiking frequency. It is not immediately clear which neuron will allow for a better reconstruction of the underlying stimulus, as the influence of the frequency and the precision of the spikes would have to be pitted against each other.

In this chapter I will derive a number of exact and approximate relations for the MSE of the optimal filters described in the previous chapter. I will provide a solution for the stationary distribution of the posterior variance of the optimal filter for the OU process, showing that this distribution diverges when the average interspike interval is longer than the relaxation time of the variance. I will present a number of approximate treatments of limiting cases, for low firing rates and for the diffusion limit of large observation noise. Finally I will provide a treatment of the average posterior kernel, which allows us to study the MSE of the optimal filter of general Gaussian processes. My goal in this chapter is to develop exact and approximate methods to study the average MSE per-

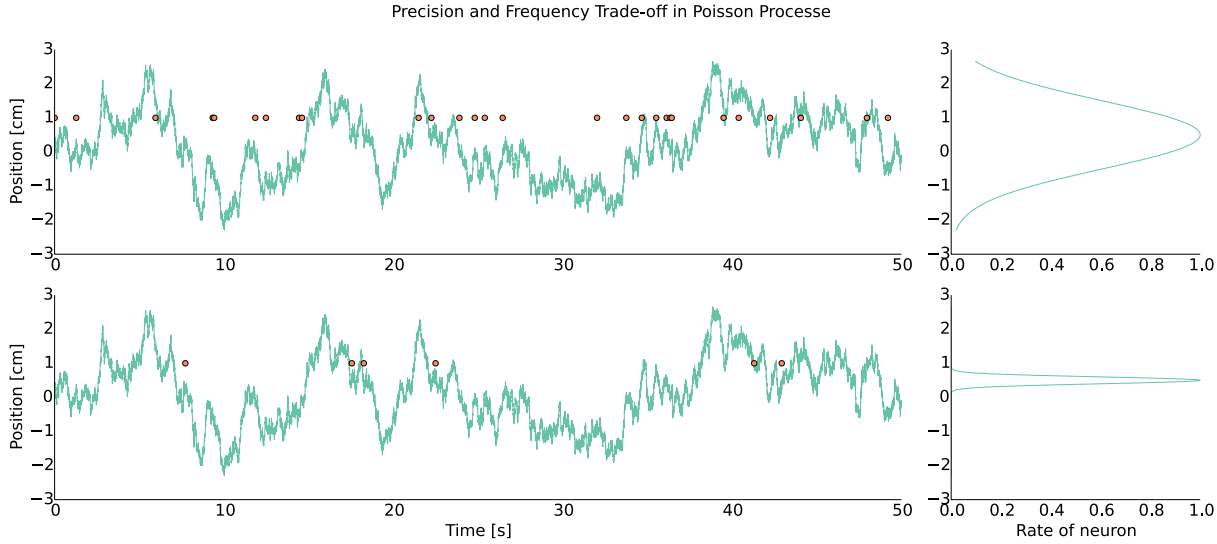


Figure 3.1: The tradeoff between precision and frequency. The upper plot shows the spike train of a single neuron with a Gaussian tuning function with unit width responding to a stochastic stimulus. There are quite a few spikes but they are fairly unreliable. The lower plot shown a neuron with a Gaussian tuning function with a width of 0.2. There are nearly no spikes, but they spike only when the neuron is in a narrow range around the neuron's tuning centre.

formance of the optimal filter of a system observed through a population of doubly stochastic Poisson processes. Armed with that, I will in chapter 5 look at the optimal strategies for these processes to encode the state of system, namely the ones that minimise the MSE.

## *3.1 The MSE for Dense Gaussian DSPP Observations*

In section 2.4 I have discussed case of dense populations of Gauss-Poisson neurons, first introduced in (Huys et al., 2007). This framework allows for a number of simplifications. Most importantly, the posterior variance of the filter obeys a SDE with drift and jumps where the jumps occur with a state-independent rate $\hat{\lambda}$. As I have discussed above, the MMSE estimator gives the optimal estimator for a data-generating distribution $P(Y|X;E,\{\theta_i\},\phi)$. However, if the response properties of the neural population change (through changes in $E$, $\phi$ or the positioning of the tuning functions), the performance of the estimator will change as well and one can ask which is the encoder that provides the lowest MMSE.

As in section 2.4, let the stimulus be a stochastic process $X(t)$, and the observations be spike trains of a dense population of Gauss-Poisson neurons $\boldsymbol{N}(t) = (N^1(t),\ldots,N^M(t))^\top$. Writing $\boldsymbol{N}_{0:t} = \{\boldsymbol{N}(v), \forall v \in [0,t]\}$, the posterior distribution is

$$P(X(t)|\boldsymbol{N}_{0:t}) = \mathcal{N}\left(\mu(t;\boldsymbol{N}_{0:t}), \Sigma(t;\boldsymbol{N}_{0:t})\right),$$

where $\mu(t;\boldsymbol{N}_{0:t})$ and $\Sigma(t;\boldsymbol{N}_{0:t})$ are solutions to equation (2.21). The MMSE matrix can be written as

$$\boldsymbol{MMSE}(t;\{\theta_i\},E) = \boldsymbol{E}_{X(t),\boldsymbol{N}_{0:t}}\left[\left(X(t)-\mu(t;\boldsymbol{N}_{0:t})\right)\left(X(t)-\mu(t;\boldsymbol{N}_{0:t})\right)^\top\right] \equiv \epsilon(t).$$

I am interested in the ensemble average over all possible realisations of both the signal as the observation processes.

Throughout this chapter, I assume that the posterior distribution used to evaluate the posterior mean estimator is the same distribution as the one that is generating the observations $\boldsymbol{N}_{0:t}$. This is called the matched situation. If complete information about the distribution $P(\boldsymbol{N}_{0:t}|X(t))$ was unavailable, one would be forced to work with an approximate posterior. In the matched case, however, I can simplify the expression for $\epsilon(t)$. Writing out the expectation over $X$ and $\boldsymbol{N}$ gives

$$\epsilon(t) = \int d\mu\left(X(t)\right) \int d\mu(\boldsymbol{N}_{[0:t]}) \left(X(t) - \mu(t;\boldsymbol{N}_{0:t})\right) \left(X(t) - \mu(t;\boldsymbol{N}_{0:t})\right)^{\top} P(X|\boldsymbol{N}_{0:t}) P(\boldsymbol{N}_{0:t}).$$

The average over $P(X|\boldsymbol{N}_{[0:t]})$ will just yield the posterior variance $\Sigma(\boldsymbol{N}_{0:t})$, leading to

$$\epsilon(t) = \boldsymbol{E}\left[\Sigma(\boldsymbol{N}_{0:t})\right] = \boldsymbol{E}\left[\Sigma(N_{0:t})\right], \tag{3.2}$$

where in the last step I have used that the posterior variance is only a function of the population spike count $N_{0:t} = \sum_i N^i_{0:t}$ and not of the full spike train $\boldsymbol{N}_{0:t} = \left(N^1_{0:t}, N^2_{0:t}, \ldots\right)^{\top}$. This makes it much simpler to treat the averages, but they still remain intractable. To get a sense of the problem, for every possible spike count, one would have to average over all possible spike times for those spikes, considering the evolution of $\Sigma$ from its initial value according to the dynamics given in equation (2.21b), and then average over all possible spike counts. This has been done for the case of static stimuli in (Yaeli and Meir, 2010). When the stimulus is static, the averages are simplified by the fact that the variance does not change between spikes. The posterior variance is given by

$$\Sigma(t) = \left(\Sigma(0)^{-1} + N(t)E^{\dagger}\right)^{-1}.$$

Averaging over the spike trains then amounts to averaging over all possible spike counts for the given time period. This leads to

$$\epsilon_{static}(t) = \sum_{k=0}^{\infty} \left(\Sigma(0)^{-1} + kE^{\dagger}\right)^{-1} \frac{(\hat{\lambda}t)^k e^{-\hat{\lambda}t}}{k!}. \tag{3.3}$$

This simplifies further when $E = \Sigma(0)$, that is, when the tuning matrix $E$ is equal to the prior variance of the static process, leading to

$$\epsilon_{static}(t) = \Sigma(0) e^{-\hat{\lambda}t} \sum_{k=0}^{\infty} \frac{(\hat{\lambda}t)^k}{(k+1)!} = \Sigma(0) \frac{1 - e^{-\hat{\lambda}t}}{\hat{\lambda}t}. \tag{3.4}$$

When the covariance is not matched to the covariance of the prior, the infinite sum has to be evaluated numerically. The

static case has been discussed extensively by Yaeli and Meir (2010), and a similar treatment of finite state continuous time systems has been considered in (Bobrowski et al., 2009).

When considering the dynamic case, though, the average can not be evaluated explicitly. I have been able to circumvent a lot of the complexity of these averages by considering the dynamics of the posterior covariance.[3] The posterior covariance evolves according to

[3] See (Susemihl et al., 2011).

$$d\Sigma(t) = (A\Sigma(t) + \Sigma(t)A^\top + H)dt + dN(t)\left[\Sigma(t^-)E^\dagger\Sigma(t^-)\left(I + E^\dagger\Sigma(t^-)\right)^{-1}\right].$$

I will treat this as a stochastic process with a linear drift

$$B(\Sigma) = A\Sigma + \Sigma A^\top + H$$

and jumps taking $\Sigma$ to $(\Sigma^{-1} + E^\dagger)^{-1}$, which occur with rate $\hat{\lambda} = \sum_i \lambda^i(x)$. As I have shown in section 2.1, the distribution over $\Sigma$ evolves according to the differential Chapman-Kolmogorov equation.[4] Taking the drift $B(\Sigma)$ and the transition probability of jumping from $\Sigma$ to $\Sigma'$

[4] Gardiner, C. W. (2004). *Handbook of Stochastic Methods: for Physics, Chemistry and the Natural Sciences*, volume Vol. 13 of *Series in synergetics*. Springer

$$W(\Sigma', \Sigma) = \hat{\lambda}\delta\left(\Sigma' - (\Sigma^{-1} + E^\dagger)^{-1}\right),$$

the differential Chapman-Kolmogorov equation becomes

$$\frac{\partial P(\Sigma, t)}{\partial t} = -\nabla\left[B(\Sigma)P(\Sigma, t)\right] + \int d\Sigma'\left(P(\Sigma', t)W(\Sigma, \Sigma') - P(\Sigma, t)W(\Sigma', \Sigma)\right),$$

leading to

$$\frac{\partial P(\Sigma, t)}{\partial t} = -\nabla\left[B(\Sigma)P(\Sigma, t)\right] + \hat{\lambda}C(\Sigma)P\left((\Sigma^{-1} - E^\dagger)^{-1}, t\right) - \hat{\lambda}P(\Sigma, t). \tag{3.5}$$

The term $C(\Sigma)$ is resultant of the integration of the Dirac delta function and is given by

$$C(\Sigma) = \frac{1}{|\det(J(\Sigma))|},$$

where $J(\Sigma)$ is

$$J(\Sigma)_{(i,j),(k,l)} = \frac{\partial(\Sigma^{-1} + E^\dagger)^{-1}_{i,j}}{\partial\Sigma_{k,l}} = (I + E^\dagger\Sigma)^{-1}_{k,i}(I + \Sigma E^\dagger)^{-1}_{j,n}.$$

I am here considering the four-index quantity $J(\Sigma)$ as a two-index matrix, by ordering the entries of the matrix $\Sigma$ into a vector. The exact order in which I do this is unimportant, as a change in the order would only amount to a change in sign of the determinant, which in turn only appears in equation (3.5) through its absolute value.

Through equation (3.5) I can study the MMSE of a neural encoder, as it is given by an average over $P(\Sigma, t)$. The evolution of the average of some function $f$ over $P(\Sigma, t)$ can be written as[5]

[5] See section 2.1.

$$\frac{\partial \boldsymbol{E}_t[f(\Sigma)]}{\partial t} = \int d\Sigma \Big[\nabla f(\Sigma) B(\Sigma) P(\Sigma,t) + \hat{\lambda} f(\Sigma)\Big(C(\Sigma) P\Big((\Sigma^{-1} - E^{\dagger})^{-1}, t\Big) - P(\Sigma,t)\Big)\Big].$$

Changing the variables in the second integral, using $\Sigma' = (\Sigma^{-1} - E^{\dagger})^{-1}$, $d\Sigma' = C(\Sigma) d\Sigma$, $\Sigma' = (\Sigma^{-1} + E^{\dagger})^{-1}$, leads to

$$\frac{\partial \boldsymbol{E}_t[f(\Sigma)]}{\partial t} = \int d\Sigma \nabla f(\Sigma) B(\Sigma) P(\Sigma,t) + \hat{\lambda} \int d\Sigma\, P(\Sigma,t)\Big(f\Big((\Sigma^{-1} + E^{\dagger})^{-1}\Big) - f(\Sigma)\Big).$$

The evolution of the MMSE is obtained by taking $f(\Sigma) = \Sigma$, yielding

$$\frac{d\epsilon(t)}{dt} = A\epsilon(t) + \epsilon(t)A^{\top} + H - \hat{\lambda} \boldsymbol{E}_T\Big[\Sigma E^{\dagger} (\Sigma^{-1} + E^{\dagger})^{-1}\Big]. \tag{3.6}$$

This is still intractable, as the far right hand term involves an average over a nonlinear function of $\Sigma$. There are many ways to deal with that approximately. One possibility is to evaluate the average numerically by sampling from the paths of $\Sigma(t)$ according to equation (2.21b). Another option is to approximate the distribution $P(\Sigma, t)$ by some parametric distribution and obtain an approximation for the evolution of $\epsilon(t)$. The simplest such parametric distribution would be a point mass at the expected value of $\Sigma(t)$ with probability 1. This is the so-called Mean-Field approach, where one simply disregards all fluctuations in $\Sigma$ and approximate all averages $\boldsymbol{E}_t[f(\Sigma)]$ by $f(\boldsymbol{E}_t[\Sigma])$. This leads to the mean-field evolution of $\epsilon(t)$

$$\frac{d\epsilon(t)}{dt} \approx A\epsilon(t) + \epsilon(t)A^{\top} + H - \hat{\lambda}\, \epsilon(t) E^{\dagger} \Big(\epsilon(t)^{-1} + E^{\dagger}\Big)^{-1}. \tag{3.7}$$

The mean-field and sampling approaches are compared in figure 2.2. As one can see, the mean-field approximation yields extremely good results, for the stationary and relaxation behavior of $\epsilon(t)$.

Although one can now in principle evaluate the temporal evolution of the MMSE, I will focus mostly on the stationary case, as it allows for some interesting insights. The treatment of the full time-dependent distribution $P(\Sigma, t)$ does not allow for analytical solutions, and since I am mainly interested in the dependence of the MMSE on the parameters of the encoder and of the statistics of the environment, which should not change significantly in the time-scales relevant to changes in $\epsilon(t)$, I will now look deeper into the stationary distribution of $\Sigma(t)$.

### *3.2 Solving for the Stationary Distribution*

Although I am interested in finding the optimal encoder, which in turn is a function of the expected variance, one can gain

a lot of insight into the nature of the encoder by studying the full distribution of posterior variances. Here I will therefore consider the distribution of variances in its steady state. Considering equation (3.5), one can write the stationary condition as

$$\nabla\left[B(s)P(s)\right] = \hat{\lambda}C(s)P\left((s^{-1}-E^{\dagger})^{-1}\right)-\hat{\lambda}P(s), \qquad (3.8)$$

*Exact Solution for the One-Dimensional Case*

I will provide an exact solution for the one-dimensional OU case. This was proposed in (Susemihl et al., 2011), and an approximate extension to the multidimensional case was presented in (Susemihl et al., 2013). The solution relies on one simple observation, which can be glanced from figure 2.2: The posterior variance never exceeds the stationary value of the unobserved variance. I will give a simple proof for the one-dimensional OU case. Taking the simple OU process given by the SDE[6]

$$dX(t) = -\gamma X(t)dt + \eta^{1/2}dW(t)$$

and considering one-dimensional tuning functions $\lambda_m$ given by

$$\lambda_m(x) = \phi\exp\left(-\frac{(x-\theta_m)^2}{2\alpha^2}\right),$$

equation (3.5) will simplify to

$$\frac{\partial P(\Sigma,t)}{\partial t} = \frac{\partial}{\partial\Sigma}\left((2\gamma\Sigma-\eta)P(\Sigma,t)\right)+\hat{\lambda}\left(\frac{\alpha^2}{\alpha^2-\Sigma}\right)^2 P\left(\frac{\alpha^2\Sigma}{\alpha^2-\Sigma},t\right)-\lambda P(\Sigma,t). \qquad (3.9)$$

The stochastic dynamics of the posterior variance $\Sigma(t)$ in this case is

$$d\Sigma(t) = -2\gamma\Sigma(t)+\eta-\frac{\Sigma(t)^2}{\Sigma(t)+\alpha^2}dN(t).$$

The stationary variance of the unobserved process is given by $\Sigma^0 = \eta/2\gamma$ and it is easy to see that if $\Sigma(t) > \eta/2\gamma$, the variation of $\Sigma(t)$ will always be $d\Sigma(t) < 0$. Therefore, one can conclude that in the stationary regime, $P(\Sigma > \eta/2\gamma) = 0$, and therefore $P(\Sigma) = 0, \Sigma > \eta/2\gamma$ for the stationary distribution. A full derivation of this result has been given in (Susemihl et al., 2013).

Thus it is established that in the stationary regime the probability of finding a variance higher than the equilibrium variance of the process $\eta/2\gamma$ will be zero. I will now look for a solution for the stationary distribution, which obeys

$$\frac{\partial}{\partial\Sigma}\left[(2\gamma\Sigma-\eta)\,P(\Sigma)\right] = \hat{\lambda}\left(\frac{\alpha^2}{\alpha^2-\Sigma}\right)^2 P\left(\frac{\alpha^2\Sigma}{\alpha^2-\Sigma}\right)-\lambda P(\Sigma). \qquad (3.10)$$

[6] Note that this is still a specific case of equation (2.4).

This is a delay-differential equation, with nonlinear delays. This is called so because the derivative of the probability at a given variance depends on the value of $P$ for other variances. These kinds of equations arise in physics, where the delay is in the time variable, and arises from some temporal restriction in the interaction of different systems. To fully specify a DDE, one must give an initial condition in an interval, so that the delayed terms are defined throughout the equation. In the case of equation (3.10), the initial condition is given by $P(\Sigma) = 0, \forall \Sigma > \eta/2\gamma$.

I will define the function

$$j(\Sigma) = \frac{\alpha^2 \Sigma}{\alpha^2 + \Sigma},$$

which gives the variance after a spike, and the intervals $S_n = (j^n(\eta/2\gamma), j^{n-1}(\eta/2\gamma)]$ where $j^0(\eta/2\gamma) = \eta/2\gamma$. Clearly, the first term on the right hand side of equation (3.10) will be zero in $S_0$, as any jumps ending there would have to originate from $s > \eta/2\gamma$, where the stationary probability is zero. The equation for the distribution in $S_0$ will be simply

$$\frac{\partial}{\partial \Sigma} \left[ (2\gamma\Sigma - \eta) P(\Sigma) \right] = \hat{\lambda} P(\Sigma).$$

One can readily see that this will be solved by

$$P(\Sigma) = C \left( \frac{\eta}{2\gamma} - \Sigma \right)^{\frac{\hat{\lambda}}{2\gamma} - 1}, \forall \Sigma \in S_0. \tag{3.11}$$

Given the result for $S_0$ one can subsequently treat the equation (3.10) in $S_1$ as a simple ordinary differential equation with a non-homogeneity given by the solution in $S_0$. This is the general approach to solving delay-differential equations, usually called the method of steps. One can then recursively solve for all subsequent intervals. Figure 3.2 shows the numerical solution for the subsequent intervals along with an histogram of the variances and the van Kampen approximation to the distribution derived below.

One particularly interesting characteristic of equation (3.11) is its exponent. The sign of the exponent in equation (3.11) depends on the specific value of $\hat{\lambda}$ and $2\gamma$. If $\hat{\lambda} > 2\gamma$, the exponent will be larger than 0, leading the distribution to tend to 0 as $s$ tends to $s_0$. If, however, $\hat{\lambda} < 2\gamma$, the exponent will be negative, leading the distribution to diverge around $s_0$. Notice that $s_0$ is the worst possible performance our encoder can achieve, as it is the stationary variance of the unobserved process. This means that whenever the firing rate of the population is below a certain value, the probability distribution of our MSE will be dominated by its worst possible value. These results are illustrated in figure 3.2. This is very interesting,

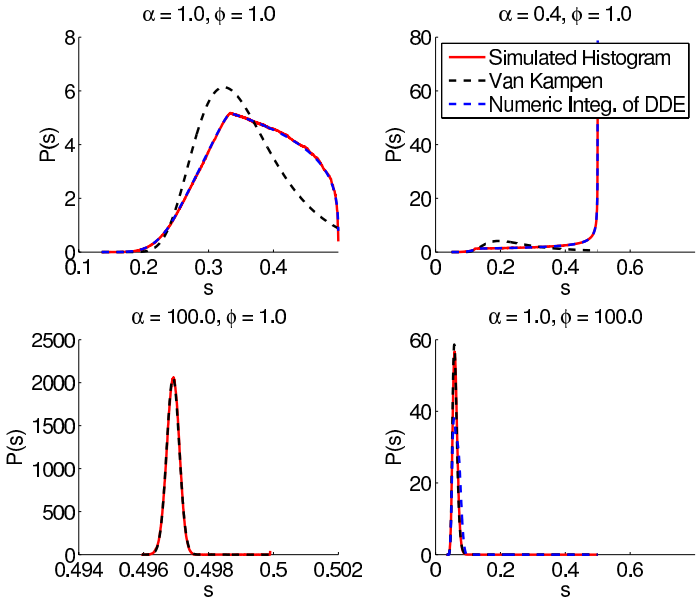


Figure 3.2: The two approaches to solving for the equilibrium distribution described, shown across a range of parameter values.

as it relates two different time scales, one $(1/2\gamma)$ describing how long information about the observed process stays relevant, the other $(1/\hat{\lambda})$ describing the average time between observations. It is intuitive to say that if the interval between spikes is much larger than the correlation time of the process, one would expect estimation of the system's state to be bad. But here I have provided a simple analytic argument showing that, for a simple system whenever the average inter spike interval is longer than the correlation time of the observed process, the mode of the distribution of errors will be the worst possible error for the estimator.

Although the obtained distribution is always valid in the interval $S_0$, it can be shown to hold in the limit of low firing rates as well. This can be extended numerically to higher dimensions. Assuming the population firing rate $\hat{\lambda} \ll 2\gamma$, one will find that the expected interspike interval is much longer than the characteristic time of the variance's dynamics. It is then safe to assume, that whenever a spike is fired, the variance is very close to the stationary variance of the unobserved process $\Sigma_0 = \eta/2\gamma$. The evolution of it after the spike time $t_s$ will then be given by

$$\Sigma(t) = e^{-2\gamma(t-t_s)}\Sigma' + \Sigma_0\left(1-e^{-2\gamma(t-t_s)}\right),$$

where $\Sigma' = j(\Sigma_0)$. Solving for the time, one obtains

$$\tau(\Sigma) \equiv (t-t_s) = -\frac{1}{2\gamma}\log\left(\frac{\Sigma_0-\Sigma}{\Sigma_0-\Sigma'}\right).$$

Clearly, if the spikes are sampled from a Poisson process, then the interspike intervals have an exponential distribution $P(\tau) \propto$

$e^{-\hat{\lambda}\tau}$. A change of variables thus leads to the density

$$P(\Sigma) = P(\tau)\left|\frac{d\tau}{d\Sigma}\right| \propto e^{-\hat{\lambda}\tau+2\gamma\tau}.$$

Inserting the definition for $\tau$ one recovers equation (3.11). This is an approximation for $P(\Sigma)$ throughout the range of $s$ for a particular parameter limit, whereas before I had derived an exact result for any parameters, but limited to a small range of values of $\Sigma$.

*An Extension to the Multidimensional Case*

I will derive a similar limit for the multidimensional case. First assume $\hat{\lambda}$ is small enough for the covariance $\Sigma$ to have relaxed to the stationary covariance of the unobserved process $\Sigma_0$. After a spike the covariance is then given by $\Sigma' = \left(\Sigma_0^{-1} + E^{\dagger}\right)^{-1}$. The evolution of $\Sigma(t)$ after a spike at $t_s$ is

$$\Sigma(\tau) = e^{\tau A}\Sigma' e^{\tau A^{\top}} + \int_0^{\tau} e^{tA}\eta e^{tA^{\top}} dt.$$

It is not possible to proceed as before, since the mapping from the matrix space of $\Sigma$ to the one-dimensional time space can not be explicitly written as above. One could try to work out the densities for individual entries of the matrix $\Sigma$ but these would possibly not be one-to-one. One alternative is to evaluate the marginals of the matrix entries numerically through integration of the dynamics of $\Sigma(\tau)$. One could thus integrate $\Sigma(t)$ numerically until it has reached the stationary value $\Sigma_0$, and evaluate the derivatives $\dot{\Sigma}$ numerically. This allows one to look at the marginal probabilities of each entry of $\Sigma$, leading for example to

$$P(\Sigma_{11}) = \frac{P(\tau(\Sigma_{11}))}{\left|\frac{d\Sigma_{11}}{d\tau}\right|} \propto \frac{e^{-\hat{\lambda}\tau}}{\left|\frac{d\Sigma_{11}}{d\tau}\right|},$$

where $\tau(\Sigma_{11})$ is simply the time associated to that particular value of $\Sigma_{11}$ in the numerical integration. If $A$ introduces interactions between the entries of the covariance matrix, however, this result will not prove as powerful. As an example, consider the Matern processes treated in (Susemihl et al., 2013), where the unobserved covariance evolves as

$$\begin{aligned}
\frac{d\Sigma_{11}}{dt} &= 2\Sigma_{12}, \\
\frac{d\Sigma_{12}}{dt} = \frac{d\Sigma_{21}}{dt} &= \Sigma_{22} - 2\gamma\Sigma_{11} - \gamma^2\Sigma_{12}, \\
\frac{d\Sigma_{22}}{dt} &= \eta - 4\gamma\Sigma_{12} - 2\gamma^2\Sigma_{22}.
\end{aligned}$$

The numerical approach for this Matern process is shown in figure 3.3, and again it coincides well with the distribution in the regime of low firing rates. It is important to note, that because of the higher-order dynamical nature of the covariance, the divergence of $\Sigma_{11}$ around its stationary value no longer dominates the distribution, as $\frac{d\Sigma_{11}}{dt}|_{\Sigma'} = 0$, leading to a second peak in the distribution around $\Sigma'_{11}$.

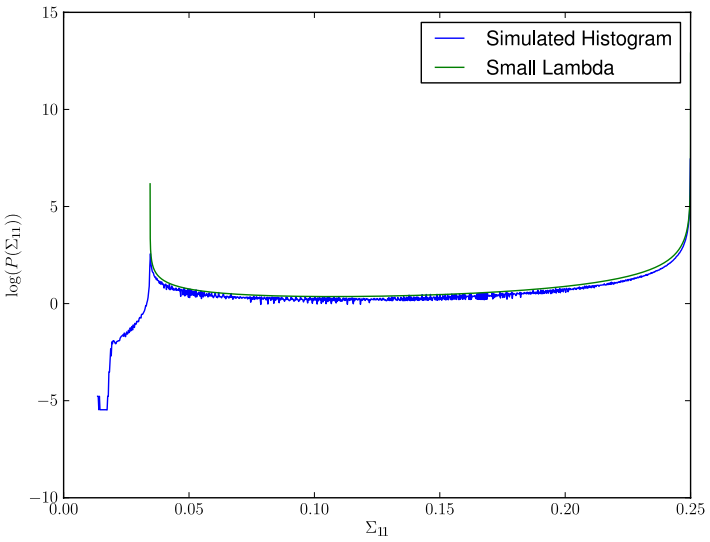


Figure 3.3: The small firing rate limit for the Matern process.

*Van Kampen Approximation*

It is interesting to consider a different limiting behaviour to gain insight into the fluctuations of $\Sigma$. The Van Kampen approximation is a system size expansion, often employed in the field of statistical physics. It consists of expanding the transition probabilities around the deterministic solution in powers of the inverse system size and solving the dynamics of the resulting Focker-Planck equation. It is not evident how to choose a system size for the problem at hand, but below I will show that it makes sense to consider the quantity $\gamma\alpha^2/\eta$ as a system size. It is simple to apply this to the OU problem by taking the scaled inverse variance $z = \frac{\eta}{\gamma\Sigma}$ instead of the variance. Note that although the change in the variance when a spike is observed is nonlinear, the change in the inverse variance is linear. This leads to the ODE

$$\frac{dz}{dt} = -\gamma z\,(2-z) \text{ and the jump condition } z(t) = z(t^-) + \frac{\eta}{\gamma\alpha^2}$$

for all times $t$ when there is a spike observed. Defining the jump size as $\delta = \frac{\eta}{\gamma\alpha^2}$, the differential Chapman-Kolmogorov

equation for $P(z,t)$ is given by

$$\frac{\partial P(z,t)}{\partial t} = \frac{\partial\left(\gamma z\left(2-z\right)P(z,t)\right)}{\partial z} + \hat{\lambda}\left[P\left(z+\delta,t\right)-P(z,t)\right]. \tag{3.12}$$

The evolution of the average of $z$ is given by

$$\frac{d\boldsymbol{E}\left[z\right]}{dt} = -\gamma\boldsymbol{E}\left[z\left(2-z\right)\right] + \hat{\lambda}\delta,$$

which gives the mean-field stationary solution of $z^* = 1 + \sqrt{1+\hat{\lambda}\delta}$. This mean-field approach can be refined by expanding the nonlinear terms in equation (3.12) around $z^*$ up to first order, which will yield a linear Focker-Planck equation, which can be readily solved. The stationary solution to the Focker-Planck equation is a Gaussian distribution

$$P_{eq}(z) = \mathcal{N}\left(\left(1+\sqrt{1+\hat{\lambda}\delta}\right), \frac{\hat{\lambda}\delta^2}{4\gamma\sqrt{1+\frac{\hat{\lambda}\delta}{\gamma}}}\right).$$

With this solution in hand, it is easy to find the distribution of $\Sigma$ by a change of variables. This approach is shown in figure 3.2 along with the numerical solution of the delayed-differential equation and the numerical simulations. Note that, again, equation (3.12) is still exact, and one can look at it to determine when the approximation is appropriate. I have Taylor expanded $P(z+\delta)$ keeping terms up to first order, and this will yield good approximations whenever $\frac{\eta}{\gamma\alpha^2}$ is small, that is, whenever the tuning width is large compared to the stationary variance of the unobserved process. Furthermore, the nonlinear term $\gamma z(2-z)$ was also linearised around the mean-field stationary value, which will only be a good approximation when $z$ has a small probability of wandering far from $z^*$.

In this case, $\frac{\gamma\alpha^2}{\eta}$ provides a system-size-like quantity for this system, giving us the order of magnitude of the fluctuations of the system. This can be understood by noting that the change in $\Sigma$ after a jump is given by $\Delta\Sigma = \frac{\Sigma^2}{\Sigma+\alpha^2}$. The jump can be treated as Gaussian noise if it is very small compared to the value of $\Sigma$, i.e.

$$\frac{\Delta\Sigma}{\Sigma} = \frac{\Sigma^2}{\Sigma(\Sigma+\alpha^2)} = \frac{\Sigma}{\alpha^2}\frac{1}{1+\frac{\Sigma}{\alpha^2}},$$

so the size of jumps relative to $\Sigma$ are of the order of $\Sigma/\alpha^2$ and can be safely treated as Gaussian noise if $\alpha^2$ is much larger than the typical value of $\Sigma$. $\Sigma$ is at most of the order of $\eta/2\gamma$ so the limit derived above makes sense. If $\gamma\alpha^2/\eta$ is very large, the fluctuations in $\Sigma$ should be small, rendering the Van

Kampen approximation precise. This is seen to be the case in the lower left panel of figure 3.2, where the distribution of $\Sigma$ is narrow and the system size is large ($\gamma\alpha^2/\eta = 10000$), leading to a good agreement of the simulations with the Van Kampen approximation. In the lower right panel, the Van Kampen also fairs very well, but there the system size is 1. In the derivation above, however, I have used $\eta/\gamma$ as an upper bound on values of $\Sigma$. As can be glanced from the histogram in the lower right of figure 3.2, the typical value of $\Sigma$ is around 0.05, due to the high firing rate. So the actual system size as argued above would be of $\approx 50$, making the Van Kampen approximation justified.

*Prediction Error*

What if I wanted to predict the value of the stimulus $X$ at a future time, for which I have no spike train information? In the absence of spikes the optimal MMSE estimator is given by the evolution of the mean and covariance with $dN^m(t) = 0$, and one would have the predictive probability $P(X(t+\delta)|\{N^m_{0:t}\})$, with $\delta > 0$. Here I am assuming the spikes are only observed up to time $t$, and I am trying to infer $X$ at some future time $t+\delta$. The mean squared error or prediction error committed when estimating future values of $X$ in that way is given by

$$\boldsymbol{PE}_\delta(t) = \boldsymbol{E}_{X,\{N^m_{0:t}\}}\Big[\big(X(t+\delta)-\mu(t+\delta;\{N^m_{0:t}\})\big)\big(X(t+\delta)-\mu(t+\delta;\{N^m_{0:t}\})\big)^\top\Big]. \tag{3.13}$$

This gives us the matrix $\boldsymbol{MSE}$ when $\delta = 0$. For $\delta > 0$ it gives the prediction error matrix. Given a value of $X(t)$ and a realisation of the Wiener process $W(s)$ for $t \le s \le t+\delta$, one has

$$X(t+\delta) = \int_0^\delta e^{-sA}H^{1/2}dW(s) + e^{-\delta A}X(t).$$

Clearly, conditioning on $X(t)$ the above average is only over the Wiener process between $t$ and $t+\delta$. The estimator $\mu(t+\delta;\{N^m_{[0,t]}\})$ is also given by $\mu(t+\delta;\{N^m_{[0,t]}\}) = e^{-\delta A}\mu(t;\{N^m_{[0,t]}\})$ in the absence of spikes. The prediction error matrix will then be given by

$$\begin{aligned}\boldsymbol{PE}_\delta(t) = \quad \boldsymbol{E}_{W,\{N^m(0:t)\}} \quad \Big[&\Big(\int_0^\delta e^{-(\delta-s)A}HdW(t+s) + e^{-\delta A}X(t) - e^{-\delta A}\mu(t))\times \\ &\Big(\int_0^\delta e^{-(\delta-u)A}HdW(t+u) + e^{-\delta A}X(t) - e^{-\delta A}\mu(t)\Big)^\top\Big].\end{aligned}$$

Since $e^{-(\delta-u)A}H$ is non-anticipating and does not depend

on $X(t)$ or $N^m(t)$, one has that (see (Gardiner, 2004))

$$\boldsymbol{E}_W\left[\int_0^\delta e^{-(\delta-u)A}HdW(t+u)\int_0^\delta e^{-(\delta-s)A}HdW(t+s))^\top\right]=\int_0^\delta e^{-(\delta-u)A}H^2e^{-(t+\delta-u)A^\top}du$$

and therefore, changing variables,

$$\boldsymbol{PE}_\delta(t)=e^{-\delta A}\epsilon(t)e^{-\delta A^\top}+\int_0^\delta e^{-sA}He^{-sA^\top}ds. \qquad (3.14)$$

This equation also describes the evolution of the variance of a linear stochastic processes,[7] and it shows us that the prediction error is a simple function of the filtering error. This is also a consequence of the Markov nature of the posterior probability. Taking a non-Markov prior process would result in a posterior probability whose parameters could not be described by a finite set of ordinary differential equations.

[7] See (Gardiner, 2004, p.106) for an example. This derivation is closely related to the derivation of the stationary variance for the OU process therein.

Figure 3.4 shows a comparison between the theoretical result in equation (3.14) and simulation results for the prediction error. One can see that the prediction error is very well described by the derived equation.

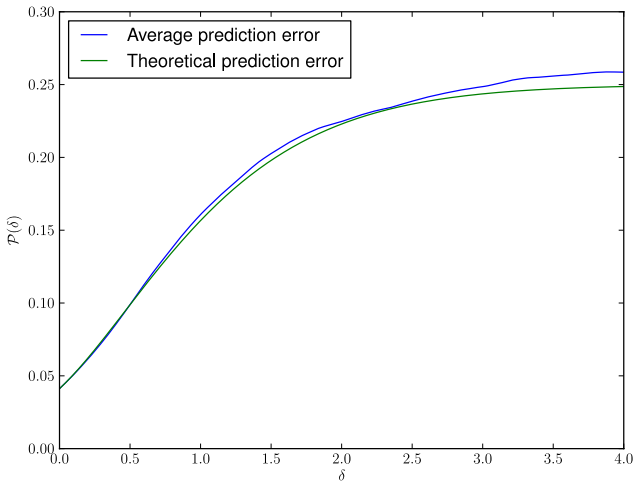


Figure 3.4: The evolution of the average prediction error $\mathscr{P}(\delta)$ is completely determined by the filtering error $\mathscr{P}(0)$. The blue line shows the prediction error obtained from the optimal filter in simulations, whereas the green line shows the evolution of the prediction error according to equation (3.14) with the initial condition given by the average filtering error obtained in the simulations. The small discrepance between both curves is due to finite sample size effects.

## *3.3 A Functional Approach to the MMSE*

In section 2.6, I have introduced Gaussian Process regression as a method of filtering general Gaussian Processes. So far, I have only considered stochastic processes which are Markovian or can be rendered Markovian by an embedding into a higher-dimensional stochastic process.[8] It is easy to adapt the treatment given in section 2.6 to the case of Poisson processes. Say I am given a Gaussian process with zero mean and covariance function $K(s,v)$ and spike trains from a dense

[8] See section 2.1

population of Gauss-Poisson neurons. Assuming up to time $t$ there have been $M$ spikes, at times $\{t_i\}$ fired by neurons $\{n_i\}$ and the tuning centres of the spiking neurons are given by $\boldsymbol{\theta} = (\theta_{n_1}, \ldots, \theta_{n_M})^\top$, the posterior mean and covariance are

$$\mu(t) = k(t, \{t_i\})^\top (G + \alpha^2 \boldsymbol{I})^{-1} \boldsymbol{\theta}, \tag{3.15}$$

and

$$\Xi(t,t) = K(t,t) - k(t, \{t_i\})^\top (G + \alpha^2 \boldsymbol{I})^{-1} k(t, \{t_j\})^\top. \tag{3.16}$$

Here $G_{i,j} = K(t_i, t_j)$ and $\alpha^2$ is the width of the tuning functions. The covariance of the posterior distribution at two points is given by

$$\Xi(s,t) = k(s,t) - \sum_{i,j} k(s,t_i) C_{ij} k(t_j, t).$$

I will call the quantity $\Xi(s,t)$ the posterior kernel, as it again defines a GP. According to the formalism derived so far the MMSE for a filtering problem is simply the expected value of the posterior kernel $\Xi(t)$ averaged over the distribution of all possible past observations. Like I have done for the MMSE, I can look into the dynamics of the posterior kernel $\Xi(s,t)$. Defining $f_t(u,v) = \Xi(t+u, t+v)$, one has

$$\frac{\partial f_t(u,v)}{\partial t} = \left( \frac{\partial}{\partial u} + \frac{\partial}{\partial v} \right) f_t(u,v).$$

It is a simple exercise in matrix inversion lemmas to show that, if a observation is obtained at time $t$ the posterior kernel will change as

$$f_t(u,v) = f_{t^-}(u,v) - \frac{f_{t^-}(u,0) f_{t^-}(0,v)}{\alpha^2 + f_{t^-}(0,0)}.$$

Taking the average over all possible observation paths one obtain the evolution of the average posterior kernel

$$\frac{\partial \boldsymbol{E}\left[f_t(u,v)\right]}{\partial t} = \left( \frac{\partial}{\partial u} + \frac{\partial}{\partial v} \right) \boldsymbol{E}\left[f_t(u,v)\right] - \hat{\lambda} \boldsymbol{E}\left[ \frac{f_t(u,0) f_t(0,v)}{\alpha^2 + f_t(0,0)} \right].$$

Again, I am most interested in the stationary case, so setting the derivative to zero one obtains

$$\left( \frac{\partial}{\partial u} + \frac{\partial}{\partial v} \right) \boldsymbol{E}\left[f(u,v)\right] = \hat{\lambda} \boldsymbol{E}\left[ \frac{f(u,0) f(0,v)}{\alpha^2 + f(0,0)} \right]. \tag{3.17}$$

Using the mean-field approximation leads to

$$\left( \frac{\partial}{\partial u} + \frac{\partial}{\partial v} \right) \boldsymbol{E}\left[f(u,v)\right] = \hat{\lambda} \frac{\boldsymbol{E}\left[f(u,0)\right] \boldsymbol{E}\left[f(0,v)\right]}{\alpha^2 + \boldsymbol{E}\left[f(0,0)\right]}. \tag{3.18}$$

This is solved by the integral equation

$$\boldsymbol{E}\left[f\left(u,v\right)\right]=k(u,v)-\frac{\hat{\lambda}}{\alpha^2+\boldsymbol{E}\left[f\left(0,0\right)\right]}\int_0^\infty \boldsymbol{E}\left[f\left(s+u,0\right)\right]\boldsymbol{E}\left[f\left(0,s+v\right)\right]ds, \tag{3.19}$$

as long as the kernel $k(u,v)$ is stationary, which implies $\partial_u k(u,v) = -\partial_v k(u,v)$.[9]

[9] A stationary kernel is such that $k(t,v) = k(\|t-v\|)$. If $t > v$ and $t > v+d$, then $k(t,v+d) = k(t-d,v)$. Differentiating with respect to $d$ we obtain the desired result.

Equation (3.19) allows one to approximate the shape of the posterior kernel directly from the prior kernel, without having to resort to the Markovian structure of the process as we had done before. This is very convenient as it allows one to treat non-Markovian GP's such as the one defined by the squared exponential or Radial Basis Function kernel.[10] This relies on the Mean-field approximation, but it is still a very pleasing result, as it additionally allows one to estimate the shape of the entire posterior kernel, not only of the one-time variance. If one is only interested in the filtering error however, it suffices to take the function $g(u) = \boldsymbol{E}\left[f\left(u,0\right)\right]$, where the filtering error is then given by $g(0)$. For that the equation simplifies to

[10] The SE or RBF kernel is given by $k(t,v) = c\exp(-|t-v|^2/L^2)$.

$$g(u)=k(u,0)-\frac{\hat{\lambda}}{\alpha^2+g(0)}\int_0^\infty g(s+u)g(s)ds \tag{3.20}$$

One way to solve equation (3.20) is to simply discretise the real line over some interval $[0,D]$ and iterate equation (3.20) numerically. I will discuss this approach shortly in appendix A.3. This is shown in figure 3.5 for the OU kernel $k_{OU}(s,t) = \exp(-|s-t|/k)$, for the Matern kernel $k_{mat}(s,t) = (1+\sqrt{3}|t-s|/k)\exp(-\sqrt{3}|t-s|/k)$ and the RBF kernel $k_{dbf} = \exp(-|t-s|^2/2k^2)$. As one moves towards smoother processes, the variance of the posterior kernel decreases, and the effect of the observations becomes more pronounced.

This can be used to study the MMSE of more complex Gaussian processes. The RBF kernel and the associated Gaussian process have been the subject of great interest, specially in the Machine Learning community, as the squared exponential form of it often allows one to simplify a number of expressions in Gaussian averages. Marc Deisenroth, for example, proposed to use the Gaussian form of the RBF kernel to average over uncertainty in the input $t$ of the process as well as in the observation $Y$.[11] However, though it has proven very useful in ML, the RBF kernel is often criticised for being too smooth.[12] A function $f(s)$ drawn from a GP with an RBF kernel is $\mathscr{C}^\infty$, leaving little room for randomness in its proper sense. On the other hand, Huys et al. (2007) have argued that experimental trajectories of freely moving animals show an autocorrelation that is compatible with the RBF kernel. Though it might be a too strong prior to impose on natural

[11] Deisenroth, M. P., Huber, M. F., and Hanebeck, U. D. (2009). Analytic moment-based gaussian process filtering. In *Proceedings of the 26th annual international conference on machine learning*, pages 225–232. ACM

[12] Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 1st edition

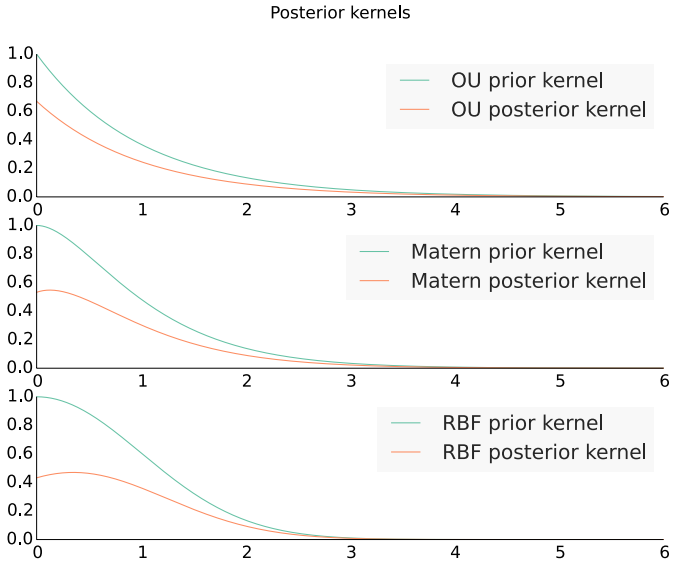


Figure 3.5: The prior and posterior kernels for the filtering problem for three classes of GP's.

stimuli, the RBF kernel might still be useful to model animal behaviour, with its longer time dependence.

## 3.4 *Alternative Performance Measures for Estimation*

I have chosen to focus on the mean-squared error of an estimation problem as a measure of efficiency of a neural population code. This is by no means the only alternative there is. Many studies in computational neuroscience have focused on the Fisher information,[13] and information-theoretical quantities such as the entropy or the mutual information of a code.[14] I will review the motivation and discuss the application of these tools in the present setting if merited.

### *Fisher Information*

The Fisher information gives an alternative measure to the amount of information carried by an observation $Y$ about an unobserved parameter or variable $x$. The Fisher information $\mathcal{I}(x;Y)$ of $Y$ about $x$ is given by

$$\mathcal{J}(x;Y) \equiv \int P(Y|x)\left(\frac{\partial \log\left(P(Y|x)\right)}{\partial x}\right)^2 dY. \qquad (3.21)$$

Intuitively, the Fisher information tells one how sensitive the likelihood of an observation is to a change in the unobserved variable $x$ at that particular value. So, a code that gives a high Fisher information on average will be very sensitive to changes in $x$, allowing for good estimation of $x$ from $Y$. The relationship between estimation and the Fisher information

[13] Ganguli, D. and Simoncelli, E. P. (2011). Implicit encoding of prior probabilities in optimal neural populations. (December 2010):6–9; Zhang, K. and Sejnowski, T. T. J. (1999). Neuronal tuning: To sharpen or broaden? *Neural Computation*, 11(1):75–84; and Brunel, N. and Nadal, J. (1998). Mutual information, Fisher information, and population coding. *Neural Computation*, 10(7)

[14] Schneidman, E., Still, S., Ii, M. J. B., and Bialek, W. (2003). Network Information and Connected Correlations. 1(December):3–6; Tkacik, G., Prentice, J. S., Balasubramanian, V., and Schneidman, E. (2010). Optimal population coding by noisy spiking neurons. *Proceedings of the National Academy of Sciences of the United States of America*, 107(32):14419–24; ; and Brunel, N. and Nadal, J.-P. (1997). Optimal tuning curves for neurons spiking as a poisson process. In *ESANN*. Citeseer

can be made rigorous through the Cramér-Rao bound. If one has an estimator $\hat{x}(Y)$ of $x$ based on observations of $Y$, the Cramér-Rao bound states that the variance of that estimator is bounded by

$$\boldsymbol{E}_{Y|x}\left[(\hat{x}(Y)-x)^2\right] \geq \frac{\left(1+\frac{\partial b(x)}{\partial x}\right)^2}{\mathscr{J}(x;Y)}, \tag{3.22}$$

where $b(x) = \boldsymbol{E}_{Y|x}[\hat{x}(Y)] - x$ is the bias of the estimator. If $\hat{x}(Y)$ is unbiased this simplifies to

$$\boldsymbol{E}_{Y|x}\left[(\hat{x}(Y)-x)^2\right] \geq \frac{1}{\mathscr{J}(x;Y)}.$$

In the multivariate case this becomes a restriction on the positive-definiteness of the MSE matrix, more precisely, for the unbiased case, one has that for any $v$ it holds that

$$v^\top \boldsymbol{E}_{Y|x}\left[(\hat{x}(Y)-x)(\hat{x}(Y)-x)^\top\right] v \geq v^\top \mathscr{J}(x;Y)^{-1} v,$$

which means that the matrix $\boldsymbol{E}_{Y|x}\left[(\hat{x}(Y)-x)(\hat{x}(Y)-x)^\top\right] - \mathscr{J}(x;Y)^{-1}$ is positive semidefinite.

The Cramér-Rao bound can also be extended to a Bayesian setting, where one is no longer estimating a fixed parameter but a random variable. Considering $X$ a random variable with distribution $P(X)$, one obtains the Bayesian Cramér-Rao Bound (BCRB),

$$\boldsymbol{E}_{X,Y}\left[(\hat{X}(Y)-X)^2\right] \geq \frac{1}{\boldsymbol{E}_X\left[\mathscr{J}(X;Y)\right] + \mathscr{I}(X)},$$

where

$$\mathscr{I}(X) = \int dX P(X) \left(\frac{\partial \log P(X)}{\partial X}\right)^2.$$

Unlike the Cramér-Rao Bound given in equation (3.22), the BCRB gives a bound on the performance of a given code in an environment regardless of the system's state.

The Fisher information is particularly popular in the neuroscience community partly because it has a convenient form for rate-based models. Suppose one has a Poisson neuron with some tuning function $f(X)$. The probability of a spike count $r$ in a time interval of duration $T$ is given by

$$P(r|X) = \frac{e^{-Tf(X)T}(Tf(X))^r}{r!}, \quad \log(P(r|X)) = r\log(Tf(X)) - Tf(X) - \log r!.$$

This leads to the Fisher information

$$\mathscr{J}_{Poiss}(X;Y) = T\frac{f'(X)^2}{f(X)} = \frac{(Tf'(X))^2}{Tf(X)},$$

which is a function of $Tf(X)$, the expected number of spikes for the experiment. In figure 3.6 I have shown the Fisher

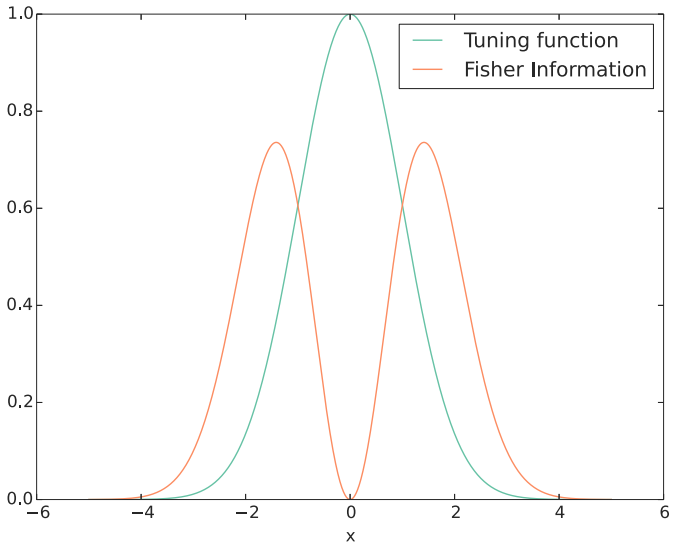


Figure 3.6: The Fisher information of a Gaussian-shaped tuning function. Note how the information is highest around the slopes of the tuning function. This approach puts a higher prize on discriminability, so the highest information is achieved when the slope of the tuning function is highest.

information for a Gaussian tuning function as a function of $X$.

One can see why this is of interest for computational neuroscientists, as rate-based models are the bread and butter of spike train analysis. A number of experiments sought to use the Fisher information as a measure of performance for coding strategies, such as (Zhang and Sejnowski, 1999; Brunel and Nadal, 1998) for example. More recently, this line of reasoning has been criticised by a number of findings. Matthias Bethge, for example, argued that the Cramér-Rao bound is loose in general and only provides a tight bound on the MMSE of a neural population code in the limit of very long times and many spikes, rendering it of limited usability.[15] For the dynamic setting I am considering, where the dynamic nature of the stimulus is central, Fisher information seems to be of little use. (Yaeli and Meir, 2010) and (Berens et al., 2011) have compared the MMSE with the Crámer-Rao bound extensively, coming repeatedly to the conclusion that the Crámer-Rao bound leads to troubling results, such as the optimal code not depending on the decoding time available. I will therefore not spend any further time investigating the Fisher information in this thesis.

[15] Bethge, M., Rotermund, D., and Pawelzik, K. R. (2002). Optimal short-term population coding: When Fisher information fails. *Neural Computation*, 14(10):2317–2351

### *Mutual Information*

The canonical tool to quantify the level of dependence between two variables in information theory is the mutual information. Though the correlation is often preferred, the mutual information provides the guarantee that it is zero if and only if the random variables are statistically independent, provid-

ing a way of robustly quantifying the level of dependence between two variables. The mutual information was defined in chapter 1 as

$$I(X;Y) = \int dX\,dY\,P(X,Y)\log\left(\frac{P(X,Y)}{P(X)P(Y)}\right),$$

which can be readily cast into

$$I(X;Y) = \int dX\,dY\,P(Y)P(X|Y)\log\left(\frac{P(X|Y)}{P(X)}\right) = \boldsymbol{E}_Y\left[H(X) - H(X|Y)\right],$$

which gives the average reduction of entropy in $X$ upon observing a random value of $Y$. One advantage of the mutual information is that it does not make any reference to an estimator or a reconstruction procedure, giving us a principled quantification of the information obtained about one variable from an observation of the other. The main disadvantage of the mutual information is that it is much harder to compute than other quantities, as one is required to estimate the whole probability distribution of $X$ and $Y$ to do so.

Another important issue one should note, is that the interpretation of the mutual information as an reduction of the entropy is more problematic in the continuous case. As I had noted before, the differential entropy can given negative values, making the last step of the equation above a bit delicate. Furthermore, the mutual information between two continuous random variables is only defined if their densities are absolutely continuous with respect to each other. In the sense of probability theory, this means that the mutual information is only defined if, whenever $P(X,Y) > 0$ then $P(X)P(Y) > 0$. The probability densities I am considering here are mostly Gaussian, and therefore positive through their whole domain, so this will not be an issue in the present case.

More recently, there have also been a number of results relating the mutual information to the MMSE of estimators. One of the most surprising results is probably is that for an additive Gaussian channel, where one is trying to estimate the value of a random variable $X$ from an observation of $Y = k^{1/2}X + N$, the following relationship holds

$$\frac{\partial I(X;Y)}{\partial k} = \frac{1}{2}MMSE(\hat{X}(Y)).$$

This holds regardless of the distribution of the random variable $X$. This is one of the so-called I-MMSE relations, which have been a very popular area of study in the field of information theory recently.[16]

For the dense Gauss-Poisson populations under consideration one can evaluate the mutual information readily. Assume that at time 0, knowledge of the system's state $X$ is given by

[16] Guo, D., Shamai, S., and Verdú, S. (2005). Mutual information and minimum mean-square error in gaussian channels. *Information Theory, IEEE Transactions on*, 51(4):1261–1282; and Merhav, N. (2011). Optimum estimation via gradients of partition functions and information measures: a statistical-mechanical perspective. *Information Theory, IEEE Transactions on*, 57(6):3887–3898

some normal distribution $P_0(X)$. One can then easily evaluate the mutual information of the system's state at time $t$, $X(t)$ and the spike train up to time $t$, $N_{0:t}$. To that end, one needs only to note that the marginal $P(X(t))$ is given by a Gaussian with moments evolving according to equation (2.19), which I will denote as $\mu^0(t)$ and $\Sigma^0(t)$. The posterior distribution is given simply by the solution of the filtering equations for the problem. The distributions are given by

$$P(X(t)) = \mathcal{N}(\mu^0(t), \Sigma^0(t)), \text{ and } P(X(t)|N_{0:t}) = \mathcal{N}(\mu(t;N_{0:t}), \Sigma(t;N_{0:t})),$$

and therefore

$$I(X(t);N_{N_{0:t}}) = \log|\Sigma^0(t)| - \boldsymbol{E}_{N_{0:t}}\left[\log|\Sigma(t;N_{0:t})|\right].$$

Here again, one is faced with an average over $P(\Sigma, t)$, this time of the logarithm of the determinant of $\Sigma(t)$. In the mean-field approximation, this will give simply the logarithm of the determinant of the matrix $\epsilon(t)$. I will show in chapter 5 that the mutual information leads to the same optimal codes as the MMSE for the OU process.

# 4
# *Optimal Control with Point Process Observations*

CLEARLY THE NERVOUS SYSTEM IS NOT SOLELY INTERESTED IN ESTIMATING THE STATE OF THE WORLD. Furthermore, if that estimate is not useful for making decisions and taking actions in a dynamic environment, there is little use for it. In the previous chapter I have discussed findings for spiking codes in an estimation context. In this chapter I will extend this approach to the framework of stochastic optimal control, and discuss how to reframe the findings in this context.

The field of optimal control has been of growing interest to the neuroscience community, but little attention has been given to the issue of optimal coding in a control context. Here I will study a simple case of linear quadratic control observed through a dense population of Gauss-Poisson neurons, for which I have been able to derive a closed-form expression for the optimal cost-to-go. This allows one to study the expected control cost in an experiment as a function of the encoder, similarly to what I have done with the MMSE in the previous chapters. Furthermore, in chapter 5 I will compare these two approaches, showing that in a couple of simple examples, the control-optimal and the MMSE-optimal encoders differ significantly.

## *4.1 Optimal Control*

The field of control theory is concerned with the steering and controlling of systems, always with the minimization of a cost (or maximization of a reward) in mind. Speaking mathematically, given a system with state $X(t) \in \mathscr{X}$, with dynamics given by

$$\dot{X}(t) = f(X(t), U(t)), \quad X(0) = x_0$$

one would like to select the control variables $U(t) \in \mathscr{U}$ in such a way as to minimize an integrated cost function over time[1]

[1] This is an additive cost function, which is itself only a specific kind of control problem. Generally one can also consider more complex cost functions as well, that depend on the minimum or maximum of the state or multiplicative cost functions.

$$C(X_{0:T}, U_{0:T}) = \int_0^T c(X(s), U(s), s)ds + h(X(T)).$$

Here $c(x, u, t)$ specifies a cost rate accumulated over time and $h(x)$ describes some final goal the system should achieve at the end of the control problem.

In a purely deterministic setting, the solution to the control problem would be a policy $U^* : \mathscr{X} \times \mathbf{R} \to \mathscr{U}$ which for each system state and time gives a control to be applied to the system when it is in that state at that time. One would have

The minimum of the future cost over the space of controls is called the value function $V(X, t)$.

$$\min_{U_{0:T}} C(X_{0:t}, U_{0:t}, 0; x_0) = C\left(X_{0:t}, U^*(X_{0:t}), 0\right) \equiv V(x_0, 0),$$

where $V(x, t)$ is usually called the optimal cost-to-go function or the value function. $V(x, t)$ quantifies the cost one is expected to incur if he controls the system optimally through the remainder of the control problem, given that the system is at state $X(t) = x$ at time $t$.

This is a very broad formulation, but one general remark can be made, though, first put forward by Richard Bellman.[2] Bellman proposed an optimality principle, which stated that if a given policy is an optimal solution to a control problem, then the policy resulting after a number of steps of that policy must still be optimal for the remaining control problem as well. This can be formulated as a mathematical equation, the so-called Bellman equation or dynamic programming equation, which states that the minimal future cost in state $X(t)$ at time $t$ is given by the minimum over $U(t)$ of the instantaneous cost plus the minimal future cost at the resulting future state $X(t + dt)$. Mathematically, we have

[2] Bellman, R. (1952). On the theory of dynamic programming. *Proceedings of the National Academy of Sciences of the United States of America*, 38(8):716

Bellman's principle of optimality

$$V(X(t), t) = \min_{U(t)} \left[c(X(t), U(t), t)dt + V(X(t + dt), t + dt)\right]. \tag{4.1}$$

Note that in general, $X(t + dt)$ will depend on $U(t)$, making the solution of the Bellman equation difficult.

In continuous time, assuming differentiability of the value function $V$ in both its, one obtains

$$V(X(t), t) = \min_{U(t)} \left[ c(X(t), U(t), t)dt + V(X(t), t) + \frac{\partial V}{\partial t}dt + \frac{\partial V}{\partial x}dX(t) \right],$$

which leads to the Hamilton-Jacobi-Bellman equation

I will abbreviate the Hamilton-Jacobi-Bellman equation as HJB equation.

$$-\frac{\partial V}{\partial t} = \min_{U(t)} \left[ c(x, U(t), t) + \frac{\partial V}{\partial X} f(x, U(t)) \right].$$

This is often more convenient to solve, as it sometimes allows for explicit minimisation over the control. The HJB equation must be solved backwards in time, with final condition $V(x, T) = h(x)$.

*Estimation and the Separation Principle*

In the previous two chapters, I have considered the problem of filtering a stochastic process from spike trains. More specifically, given a signal, I was looking for the optimal set of parameters $\varphi^*$ for a population of neurons that minimise the MMSE of the filtering problem. Here I would like to establish a similar approach to control problems. That is, in the same sense of before, I have a noisy system observed through spike trains of a population of neurons specified by some parameters $\varphi$, but now I am concerned with controlling this noisy system. Given a cost function, I would like to determine the parameters $\varphi^*$ that minimise the control costs, instead of the filtering error. If one is interested in controlling a system, say a limb performing a movement, one must now deal with the uncertainties in the system and control it according to noisy estimates of its state. The certainty equivalence property (CEP) holds if a system one only has partial information about can be controlled ignoring the uncertainty in its state and acting as if it were fully observed. I will elaborate below.

Consider a deterministic system

$$\dot{X} = f(X(t), U(t)),$$

where I have only partial knowledge about the system's state through an initial distribution $P_0(x)$ and noisy observations $Y(t)$ of the system's state. If the certainty equivalence property holds for this system, the optimal control for the partially-observed system, will be the optimal policy of the fully observed problem applied to the mean estimate of the system's state. To be more precise, let me define the cost for the partially-observed system as

$$C(P_0, Y_{0:T}, U, 0) = \int_0^T \boldsymbol{E}\left[c(X(s), U(s), s) \mid Y_{0:s}, P_0\right] ds + \boldsymbol{E}\left[h(X(T)) \mid Y_{0:T}, P_0\right].$$

The optimal control $U^*$ will now be a function of the observations $Y_{0:t}$ up to time $t$ and the initial distribution $P_0$. If the optimal control for the fully-observed system is given by $U^*_{obs}(x,t)$, the certainty equivalence property holds if the optimal control for the partially observable process is given by

$$U^*_{part}(Y_{0:t}, P_0, t) = U^*_{obs}\left(\boldsymbol{E}\left[X(t) \mid Y_{0:t}, P_0\right], t\right).$$

This means that the uncertainty in the system's state can be treated in two independent steps, first estimating the system's state through the posterior mean and then applying the control as if our estimate of the state was certain. Hence the name certainty equivalence, as one applies the control as if they were certain about the system's state. The separation property is also frequently discussed in the literature, and it is a

stronger version of the certainty equivalence property, where the control $U^*_{obs}(x,t)$ being employed does not need to be the optimal policy for the fully observed problem, but can be related to some other control problem with full informations.

One can now ask what is the encoder that minimises the expected control costs. It is tempting to conclude from the CEP that the encoder that minimises the MMSE also minimises the control costs. This is not true, however, as I will show in chapter 5. I will now consider the case of stochastic optimal control, and then turn to the case of partially-observable stochastic optimal control. This can be treated for the case of dense Gauss-Poisson observations, and I will derive a novel relation for the optimal cost-to-go for that case.

### *4.2 Stochastic Optimal Control*

The world is a noisy place, and if to control real-world systems, one must be able to account for noise in the systems as well. One simple way to include noise is to generalise the system dynamics to a stochastic differential equation. Consider

$$dX(t) = f(X(t),U(t))dt + H^{1/2}dW(t),$$

where $W(t)$ is a standard Wiener process. It is not possible to predict the evolution of $X(t)$ exactly anymore, so one must redefine the cost function. The natural way to do so is to define it as the average over future states conditioned on the current state $X(t)$ and the controls to be applied $U(t)$. This will lead to

$$C(X,U) = \boldsymbol{E}\left[\int_0^T c(X(t),U(t),t)dt \;\middle|\; X(0),U_{0:T}\right].$$

One should mention that there are other ways to deal with the stochastic nature of the problem,[3] such as the risk-sensitive control approach, where one considers the cost function

[3] Whittle, P. (1981). Risk-sensitive linear/quadratic/gaussian control. *Advances in Applied Probability*, pages 764–777

$$C_\theta(X,U) = \frac{1}{\theta}\log \boldsymbol{E}\left[\exp\left(-\theta\left(\int_0^T c(X(t),U(t),t)dt + h(X(T))\right)\right)\right].$$

In the limit $\theta \to 0$ one recovers the former formalism. This allows one to consider risk-averse or risk-seeking control policies. I will not, however, consider this approach here.

The Bellman equation can then be extended to the stochastic case as

$$V(x,t) = \min_{U(t)} \boldsymbol{E}\left[c\left(X(t),U(t),t\right)dt + V\left(X(t+dt),t+dt\right) \mid X(t) = x, U_{[t,T]}\right]. \tag{4.2}$$

Using Itō's lemma for the variation of $V$, and averaging over the Brownian motion $dW(t)$ leads to

$$V(x,t) = \min_{U(t)}\left[c(x,U(t),t)dt + V(x,t) + \left(\frac{\partial V}{\partial t}dt + f(x,U(t))^\top \frac{\partial V}{\partial x} + \frac{1}{2}\mathrm{Tr}\left[H\frac{\partial^2 V}{\partial x^2}\right]\right)dt\right].$$

This leads to the stochastic HJB equation

$$-\frac{\partial V}{\partial t} = \frac{1}{2}\mathrm{Tr}\left[H\frac{\partial^2 V}{\partial x^2}\right] + \min_u\left[c(x,u,t) + f(x,u)^\top \frac{\partial V}{\partial x}\right]. \tag{4.3}$$

One could also consider a Poisson process as a noise source. If one takes, for example, a Poisson counting process $N(t)$, with time- and/or state-dependent rate $\lambda(X(t),t)$, and takes the system dynamics to be given by a drift-diffusion process with state-dependent jumps $j(X(t),t)$, occurring with rate $\lambda(X(t),t)$ then the SDE for the state would be,

$$dX(t) = f(X(t),U(t))dt + H^{1/2}dW(t) + j(X(t),t)dN(t).$$

This would lead to the full HJB equation for a drift-jump-diffusion process controlled by some control process $U(t)$

$$-\frac{\partial V}{\partial t} = \min_u\left[c(x,u,t) + f(x,u)^\top \frac{\partial V}{\partial x} + \frac{1}{2}\mathrm{Tr}\left[H\frac{\partial^2 V}{\partial x^2}\right] + \lambda(x,t)\left[V(x+j(x,t),t) - V(x,t)\right]\right], \tag{4.4}$$

now including the terms regarding the jump process.[4] Note that the statistics of the posterior distribution of the filtering problem from the previous chapters fit this description, namely they are a jump-drift processes with no diffusion. I will use this formalism to derive a belief state formulation of a control problem with dense Gauss-Poisson observations.

[4] Theodorou, E. and Todorov, E. (2012). Stochastic optimal control for nonlinear markov jump diffusion processes. *American Control Conference (ACC), 2012*; and Sennewald, K. and Wälde, K. (2006). "Ito's Lemma" and the Bellman equation for Poisson processes: An applied view. *Journal of Economics*, 89(1):1–36

*Linear-Quadratic-Gaussian Control*

The Linear-Quadratic-Gaussian[5] control problem is defined by linear dynamics in both the state and the control variable, a quadratic cost rate function $c$ in both the state and control and a Gaussian noise source. I will treat this problem here to illustrate the optimal control formalism. This would mean that the evolution of the state is given by the SDE

[5] LQG

$$dX(t) = (AX(t) + BU(t))\,dt + H^{1/2}dW(t), \tag{4.5}$$

where $W(t)$ is a Wiener process. Taking a cost rate given by

$$c(X(t),U(t),t) = U(t)^\top R(t)U(t) + X(t)^\top Q(t)X(t),$$

and a final cost given by $h(X(T)) = X(T)^\top Q_T X(T)$, one can solve for the value function explicitly, using the HJB equation. The HJB equation in this case will be given by

$$-\frac{\partial V}{\partial t} = \min_{U(t)}\left[U(t)^\top R(t)U(t) + x^\top Q(t)x + \frac{\partial V}{\partial x}^\top (Ax + BU(t)) + \frac{1}{2}\mathrm{Tr}\left(H\frac{\partial^2 V}{\partial x^2}\right)\right].$$

One can minimize the right hand side explicitly and eliminate $U$ from the equation. One obtains that the optimal control is given by

$$U^*(x,t) = -\frac{1}{2}R(t)^{-1}B^\top \frac{\partial V}{\partial x}\Big|_{x,t}.$$

Inserting into the HJB equation once more leads to

$$-\frac{\partial V}{\partial t} = x^\top Q(t)x + \frac{\partial V}{\partial x}^\top Ax - \frac{\partial V}{\partial x}^\top BR(t)^{-1}B^\top \frac{\partial V}{\partial x} + \frac{1}{2}\operatorname{Tr}\left(H\frac{\partial^2 V}{\partial x^2}\right). \tag{4.6}$$

It can be shown that $V$ can only have a quadratic dependence in $X$, since at the final time the cost is given by $h(X(N))$ which is quadratic, and the HJB equation will preserve this property. I will assume it is of the form $V(x,t) = x^\top S(t)x + \alpha(t)^\top x + k(t)$. Inserting this into equation (4.6) gives the ODE's for the parameters of the value function

$$\begin{aligned} -\dot{S} &= Q(t) + A^\top S(t) + S(t)A - S(t)BR(t)^{-1}BS(t), \\ -\dot{\alpha} &= A^\top \alpha(t) - S(t)BR(t)^{-1}B^\top \alpha(t), \\ -\dot{k} &= \operatorname{Tr}(HS(t)) - \alpha(t)^\top BR(t)^{-1}B^\top \alpha(t), \end{aligned}$$

with the terminal conditions $S(T) = Q_T$, $\alpha(T) = 0$ and $k(T) = 0$. The $X$-independent term $k(t)$ accounts for the future uncertainty in $X$, decreasing to 0 over time as we approach the final time $T$. Furthermore, the differential equation for $S(t)$ is a special case of the Riccati equation. The full optimal control for the LQG control problem will therefore be given by

$$U^*(x,t) = -R(t)^{-1}B^\top S(t)x.$$

These results can also be extended to the case of control- and state-dependent diffusion noise, affine dynamics and some other cases.[6]

[6] Kappen, H. (2011). Optimal control theory and the linear Bellman equation. In Barber, D., Cemgil, A. T., and Chiappa, S., editors, *Bayesian Time Series Models*, chapter 1, pages 1–31. Cambridge university press, Cambridge, 1st edition

### *4.3 Partially Observable Processes*

In general, one does not have access to the exact state of the system, and it is useful to consider cases where one is only given noisy observations of the state, as were considered in the previous chapters. The most commonly considered case of partially observable control problem is a LQG problem observed through a second diffusion process. Suppose one has as above a system $X(t)$ evolving according to equation (4.5), but instead of observing $X(t)$ directly, one observes the process $Y(t)$, which I shall call the observation process, given by

$$dY(t) = CX(t)dt + D^{1/2}dV(t). \tag{4.7}$$

Given a control trajectory $\{U(s), s \in [0,t]\}$, the problem of estimating $X(t)$ given observations $Y(s), s \in [0,t]$, is a simple

filtering problem, and is solved exactly by the Kalman-Bucy filter.[7] It will lead to a Gaussian estimate of $X(t)$ with mean $\mu(t)$ and variance $\Sigma(t)$, where $\mu$ and $\Sigma$ evolve according to

[7] See chapter 2.

$$d\mu(t) = (A\mu(t) + BU(t))dt + \Sigma(t)C^t D^{-1}\left(dY(t) - C\mu(t)dt\right), \tag{4.8a}$$

and

$$\frac{d\Sigma}{dt} = A\Sigma + \Sigma A^\top + H - \Sigma C^\top D^{-1} C\Sigma. \tag{4.8b}$$

Since in this case we do not have perfect information on the process to be controlled, we have to settle for the goal of minimizing the expected cost given our observation. Therefore, the cost to be minimized is

$$C(U_{0:T}; \mu_0, \Sigma_0) = \boldsymbol{E}\left[\int_{t_0}^{T} c(X(t), U(t), t)dt + h(X(t))\right],$$

where the average is over all future paths of $X(t)$ and all observation paths $Y(t)$. There is no analogous to the HJB equation for the incomplete information case, but I will reformulate the problem as a control problem over the belief states, that is, the state of the world as one is led to believe it is distributed given the previous observations. In the case I am discussing, the belief state is the distribution over the state variable, given by the Gaussian distribution $\mathcal{N}(\mu(t), \Sigma(t))$. The dynamics of the belief state is then given by equations equation (4.8a) and equation (4.8b). Note that when one chooses to describe the system in terms of the mean and variance of the posterior distribution, the noise process $dW(t)$ does not enter into the analysis anymore, and the observation process $dY(t)$ takes the role of the noise process. We need, however, to redefine the cost function $c(X(t), U(t), t)$ to fully specify the problem. The average cost is

The belief state is a description of an system with incomplete information which eschews describing the actual state of the system, instead describing the distribution over states. A general formulation is described in (Bertsekas, 2012).

$$\boldsymbol{E}\left[c(X(t), U(t), t)\middle|\mu(t), \Sigma(t)\right] = U(t)^\top R(t)U(t) + \left(\mu(t)^\top Q(t)\mu(t) + \mathrm{Tr}\left(Q(t)\Sigma(t)\right)\right),$$

from which one can define a belief-state cost rate, which makes no mention of the underlying unobservable process

$$c(\mu, \Sigma, U, t) = U(t)^\top R(t)U(t) + \mu(t)^\top Q(t)\mu(t) + \mathrm{Tr}\left(Q(t)\Sigma(t)\right).$$

One can now write the HJB equation for the system described by equation (4.8), leading to

$$V(\mu(t), \Sigma(t), t) =$$

$$\min_{U(t)} \boldsymbol{E}\left[U(t)^\top R(t)U(t) + \left(\mu(t)^\top Q(t)\mu(t) + \mathrm{Tr}\left(Q(t)\Sigma(t)\right)\right) + V(\mu(t+dt), \Sigma(t+dt), t+dt)\right],$$

where the expectation is now with respect to the observation process $Y(t)$. Taking the variation of $V$ with infinitesimal

time increments via Itō's lemma one has

$$dV = \frac{\partial V}{\partial t}dt + \frac{\partial V}{\partial \mu}^{\top} d\mu + \mathrm{Tr}\Big[\frac{\partial V}{\partial \Sigma}d\Sigma\Big] + \frac{1}{2}\mathrm{Tr}\Big[(\Sigma C^{\top}D^{-1}C\Sigma)_{i,j}\frac{\partial^2 V}{\partial \mu^2}\Big],$$

which leads to the HJB equation

$$\begin{aligned}-\frac{\partial V}{\partial t} = \ & \min_{U(t)} \boldsymbol{E}\Big[U(t)^{\top}R(t)U(t) + \mu(t)^{\top}Q(t)\mu(t) + \mathrm{Tr}\left(Q(t)\Sigma(t)\right) + \tfrac{\partial V}{\partial \mu}d\mu^{\top}\Big] \\ & + \mathrm{Tr}\Big[\tfrac{\partial V}{\partial \Sigma}d\Sigma\Big] + \tfrac{1}{2}\mathrm{Tr}\Big[(\Sigma C^{\top}D^{-1}C\Sigma)_{i,j}\tfrac{\partial^2 V}{\partial \mu^2}\Big].\end{aligned}$$

Minimization with respect to $U(t)$ leads to $U^*(t) = -R(t)^{-1}B^{\top}\frac{\partial V}{\partial \mu}$, which results in

$$\begin{aligned}-\frac{\partial V}{\partial t} = \ & \mu^{\top}Q(t)\mu + \tfrac{\partial V}{\partial \mu}^{\top}BR(t)^{-1}B^{\top}\tfrac{\partial V}{\partial \mu} + \mathrm{Tr}\left(Q(t)\Sigma(t)\right) + \tfrac{\partial V}{\partial \mu}d\mu^{\top} \\ & + \mathrm{Tr}\Big[\tfrac{\partial V}{\partial \Sigma}d\Sigma\Big] + \tfrac{1}{2}\mathrm{Tr}\Big[(\Sigma C^{\top}D^{-1}C\Sigma)_{i,j}\tfrac{\partial^2 V}{\partial \mu^2}\Big]. \qquad (4.9)\end{aligned}$$

This would now have to be solved backwards from $V(\mu, \Sigma, T) = \mu^{\top}Q_T\mu + \mathrm{Tr}[\Sigma Q_T]$. Equation (4.9) provides a clean formulation of the control problem in terms of the belief state, where the underlying process has been integrated over completely. This is a very useful approach and I will leverage it for the case of Point processes observations below. If I write the value function as $V(\mu, \Sigma, t) = \mu^{\top}S(t)\mu + f(\Sigma, t)$, I will obtain the same Riccati equation for $S(t)$ as in the fully observed case. Using this form for the value function, one immediately recovers the optimal control $U^*(t) = -R^{-1}(t)B^{\top}S(t)\mu$, which shows the certainty equivalence property for this system.

### *4.4 Partially Observable Processes with Poisson Observations*

Similarly to the case just discussed, we can consider the case of a stochastic system observed through a population of densely tuned Poisson processes with Gaussian tuning functions. The dynamics of the system would be the same as equation (4.5), but the observation processes would be given by a set of $M$ Poisson processes $N^m$ with rates given by

$$\lambda^m(X(t)) = \lambda \exp\Big[-\frac{1}{2}(\theta_m - X(t))^{\top}E^{\dagger}(\theta_m - X(t))\Big], \qquad (4.10)$$

where the tuning centres $\theta_m$ are positioned in such a way that the overall firing rate of the population $\hat{\lambda} = \sum_m \lambda^m(X(t))$ is independent of the system's state $X(t)$. As we have shown in chapter 2, the estimation problem is solved by the point-process analog of the Kalman-Bucy filter, first derived by Donald Snyder.[8] In the present case, with Gaussian tuning func-

[8] Snyder, D. L. (1972). Filtering and detection for doubly stochastic Poisson processes. *IEEE Transactions on Information Theory*, 18(1):91–102; and Bobrowski, O., Meir, R., and Eldar, Y. C. (2009). Bayesian filtering in spiking neural networks: noise, adaptation, and multisensory integration. *Neural computation*, 21(5):1277–320

tions, the filtering equations are

$$\begin{aligned} d\mu(t) = & \quad (A\mu(t) + BX(t))dt \\ & + \sum_i \left[\Sigma(t^-)\left(I + E^\dagger \Sigma(t^-)\right)^{-1} E^\dagger \left(\theta_i - \mu(t^-)\right)\right] dN^i(t) \end{aligned} \tag{4.11a}$$

and

$$\begin{aligned} d\Sigma(t) = & \quad \left(A\Sigma(t) + \Sigma(t)A^\top + H\right)dt \\ & - \left[\Sigma(t^-)E^\dagger \Sigma(t^-)\left(I + E^\dagger \Sigma(t^-)\right)^{-1}\right] dN(t), \end{aligned} \tag{4.11b}$$

where $dN(t) = \sum_m dN^m(t)$. I will define

$$\delta\mu(t) \equiv (A\mu(t) + BX(t))dt$$

as the continuous part of $d\mu(t)$ and

$$\Delta^i \mu(t) \equiv dN^i(t)\left[\Sigma(t^-)\left(I + E^\dagger \Sigma(t^-)\right)^{-1} E^\dagger \left(\theta_i - \mu(t^-)\right)\right]$$

as the jump part of $d\mu(t)$. Likewise, for the variance, define

$$\delta\Sigma(t) \equiv (A\Sigma(t) + \Sigma(t)A^\top + H)dt$$

and

$$\Delta\Sigma(t) \equiv dN(t)\left[\Sigma(t^-)E^\dagger \Sigma(t^-)\left(I + E^\dagger \Sigma(t^-)\right)^{-1}\right].$$

These give the evolution of the optimal Bayesian filter, with the posterior distribution over $X(t)$ conditioned on the observations $\{N^m(s), m \in [1,\dots,M], s \in [t_0, t]\}$, given by the normal distribution $\mathcal{N}(X(t); \mu(t), \Sigma(t))$. Assuming one is trying to minimize a cost given by the same cost rate $c(X(t), U(t), t)$ as before, one canwrite out the infinitesimal Bellman equation for this case as well. Since the dynamics of the system and the observations is Markov, I can use the posterior distribution as a sufficient statistic for the knowledge of the system's state. I will therefore take the belief state to be the mean and variance of the posterior distribution as before.[9] Similarly to the previous sections, I will consider the processes $N^m(s), s \leq t$ as noise to be averaged over in the future. This leads to

[9] See (Bertsekas, 2012) for a more detailed discussion.

$$V(\mu(t), \Sigma(t), t) = \min_{U(t)} \left\{ \boldsymbol{E}_{X(t)} \left[c(X(t), U(t), t)\right] + \boldsymbol{E}_{\{N^m(t)\}} \left[V(\mu(t+dt), \Sigma(t+dt), t+dt)\right] \right\}$$

According to Itō's lemma, one obtains

$$\begin{aligned} V(\mu(t+dt), \Sigma(t+dt), t+dt) = & \quad V(\mu(t), \Sigma(t), t) + \tfrac{\partial V}{\partial t} dt + \tfrac{\partial V}{\partial \mu} \delta\mu(t) + \mathrm{Tr}\left[\tfrac{\partial V}{\partial \Sigma} \delta\Sigma(t)\right] \\ & + \sum_m dN^m(t) \left[V\left(\mu(t) + \Delta^m \mu(t), \Sigma(t) + \Delta\Sigma(t), t\right) - V(\mu(t), \Sigma(t), t)\right]. \end{aligned}$$

The expectation over the noise process $N^m(t)$ in the Bellman equation can then be written as

$$\begin{aligned}\boldsymbol{E}(V_{t+dt})_{N^m(t)} = & \quad V(t) + \tfrac{\partial V}{\partial t}dt + \tfrac{\partial V}{\partial \mu}\delta\mu(t) + \mathrm{Tr}\big[\tfrac{\partial V}{\partial \Sigma}\delta\Sigma(t)\big] \\ & + \sum_m \boldsymbol{E}_{N^m(t)}\left[dN^m(t)\left[V\left(\mu(t)+\Delta^m\mu(t),\Sigma(t)+\Delta\Sigma(t),t\right)-V(\mu(t),\Sigma(t),t)\right]\right] \\ = & \quad V(t) + \tfrac{\partial V}{\partial t}dt + \tfrac{\partial V}{\partial \mu}^\top\delta\mu(t) + \mathrm{Tr}\big[\tfrac{\partial V}{\partial \Sigma}\delta\Sigma\big] \\ & + \sum_m \boldsymbol{E}_{X(t)}[\lambda^m(X(t))]\left[V\left(\mu(t)+\Delta^m\mu(t),\Sigma(t)+\Delta\Sigma(t),t\right)-V(\mu(t),\Sigma(t),t)\right],\end{aligned}$$

leading to the HJB equation

$$\begin{aligned}-\frac{\partial V}{\partial t} = & \ \mu^\top Q(t)\mu + \mathrm{Tr}\left(Q(t)\Sigma\right) + (U(t))^\top R(t)U(t) + \tfrac{\partial V}{\partial \mu}^\top\delta\mu + \mathrm{Tr}\big[\tfrac{\partial V}{\partial \Sigma}\delta\Sigma\big] \quad (4.12)\\ & + \textstyle\sum_m \boldsymbol{E}_{X(t)}\left[\lambda^m(X(t))\right]\left[V\left(\mu(t)+\Delta^m\mu(t),\Sigma(t)+\Delta\Sigma(t),t\right)-V(\mu(t),\Sigma(t),t)\right].\end{aligned}$$

Minimisation with respect to the control, gives us the optimal control policy

$$U^*(t) = -R(t)^{-1}B^\top\frac{\partial V}{\partial \mu}\bigg|_{\mu=\mu(t),\Sigma=\Sigma(t)}.$$

This yields

$$\begin{aligned}-\frac{\partial V}{\partial t} = & \quad \mu^\top Q(t)\mu + \mathrm{Tr}\left(Q(t)\Sigma\right) - \tfrac{\partial V}{\partial \mu}^\top BR(t)^{-1}B^\top\tfrac{\partial V}{\partial \mu} + \tfrac{\partial V}{\partial \mu}A\mu \quad (4.13)\\ + \mathrm{Tr}\left[\frac{\partial V}{\partial \Sigma}\delta\Sigma\right] & + \textstyle\sum_m \boldsymbol{E}_{X(t)}\left[\lambda^m(X(t))\right]\left[V\left(\mu(t)+\Delta^m\mu(t),\Sigma(t)+\Delta\Sigma(t),t\right)-V(\mu(t),\Sigma(t),t)\right].\end{aligned}$$

It can be shown that the optimal cost-to-go function is of form $V(\mu,\Sigma,t) = \mu^\top S(t)\mu + f(\Sigma,t)$, since it is of this form at the final time $T$ because of the final cost $h(x) = x^\top Q_T x$. I can now write down the equations for $S(t)$ and $f(\Sigma,t)$. The equation for $S(t)$ is the same as for the LQG case

$$-\dot{S}(t) = Q(t) - S(t)BR(t)^{-1}B^\top S(t) + S(t)A + A^\top S(t). \quad (4.14)$$

The equation for $f(\Sigma,t)$ can be shown to be[10]

[10] See appendix A.4.

$$-\frac{\partial f}{\partial t} = \mathrm{Tr}\left(Q(t)\Sigma\right) + \frac{\partial f}{\partial \Sigma}\left(A\Sigma + \Sigma A^\top + H\right) + \hat{\lambda}\Big[f(\Sigma+\Delta\Sigma,t) - f(\Sigma,t) + \mathrm{Tr}\Big(\Sigma S(t)\Sigma\left(\Sigma+E\right)^{-1}\Big)\Big]. \quad (4.15)$$

Equation (4.15) gives the contribution of the uncertainty of the estimate to the future costs. This allows one to quantify the effect of our encoder on the control costs. In chapter 5 I will use $f$ to determine the optimal encoding strategies for a simple control problem. Equation (4.15) can be shown to be solved by

$$f(\Sigma,t) = \mathrm{Tr}\left(\Sigma(t)S(t)\right) + \int_t^T \mathrm{Tr}\left(HS(u)\right)du + \int_t^T \mathrm{Tr}\Big(S(u)B^\top R(u)^{-1}BS(u)\boldsymbol{E}\left[\Sigma(u)\mid\Sigma(t)=\Sigma\right]\Big)du. \quad (4.16)$$

where the expectation is over all paths of equation (4.11b) with initial condition $\Sigma(t) = \Sigma$. I provide a derivation of this result based on the Feynman-Kac formula in appendix A.5. This equation allows one to separate the different ways in which the uncertainty affects the expected future cost. The first term accounts for the uncertainty in the present estimate of the system's state. The second term is due to the stochastic nature of the stimulus $X(t)$, and describes the accumulation of uncertainty due to the Brownian noise in that process. The third term accounts for the effect of the uncertainty on the applied control. If one is uncertain of the system's state, the control applied will not be exactly the optimal for the system's state, and additional costs will be incurred because of that. The third term is also the only one that depends on the parameters of the encoder, more specifically it depends on the future dynamics of the posterior covariance $\Sigma(t)$, which in turn depends on the firing rates and the tuning widths. A similar relation can be derived for the LQG case as well,[11] but the full result for the partially observable control with Point process observations is novel.

[11] See (Åström, 2006, p. 290) for the full derivation.

From the derivation above, it follows that the optimal control is again given by

$$U^*(t) = -R(t)^{-1}B^\top S(t)\mu(t),$$

showing that the certainty equivalence property holds in this case as well. I will discuss these issues further in section 5.4.

The finding that the certainty equivalence property holds in this simple set up, along with the exact expression for the optimal cost-to-go has not been shown in the literature to the best of my knowledge, and I believe it to provide a good starting point for the study of optimal codes in a control-theoretical setting.

# 5
# *Optimal Population Coding Revisited*

In chapter 1 I have argued that the use of the MMSE in a filtering problem is an appropriate measure for the quality of a neural encoder in a dynamic setting. In this chapter I will consider a neural population as an encoder, seeking the tuning functions for that population that minimise the decoding error. I will focus especially on the case of dense populations of Gauss-Poisson neurons. For these populations I will consider the class of linear stochastic processes described in chapter 2 and chapter 3 and investigate the dependence of the MMSE for those processes on the encoder's parameters. In the case of dense populations of Gauss-Poisson neurons I will argue that it makes the most sense to consider the width of the tuning functions as the central parameter of the encoder. I have found that for this type of population of neurons there is a finite optimal tuning width which minimises the MMSE of the filtering problem.

Following the investigation, I also present an analysis of dense populations of Gauss-Poisson neurons coding for more complex stochastic processes, such as bistable processes. In this setting, the filtering equations cease to be Gaussian, and one is forced to use approximate filtering to obtain estimates of the MMSE. I have used both the ADF approach with a Gaussian density and a simple particle filter and show results for both cases, which also show that a finite tuning width minimises the MMSE.

Finally, I will discuss some results comparing optimal population codes for filtering and control. In a set of examples, the optimal encoders for an estimation and the associated control problem are different. This is the first result of this kind to the best of my knowledge.

There has been a lot of interest in the relation between the coding mechanisms in sensory systems and the statistics of the natural environment these systems operate in. A number of studies have shown that one can obtain coding strategies similar to the ones employed by sensory systems as optimal codes for a natural ensemble of stimuli. For example, by optimising linear filters for reconstruction of naturalistic visual

stimuli under sparsity constraints, Olshausen and Field have obtained a set of filters which resemble the receptive fields of V1 pyramidal neurons.[1] Though the analysis presented here is somewhat simplistic, I believe it provides a solid foundation for similar studies of reconstruction-based analysis of optimal codes. Similar approaches have been used in the literature, but there the focus was usually on static stimuli.[2] A notable exception is the work of (Bobrowski et al., 2009), where a formalism of full-spike train decoding was presented for finite-state dynamic systems. In a certain sense, this work is an extension thereof to the case of continuous stimuli in continuous time. The present approach also makes the study of control problems a natural extension. The issue of optimal coding for control problems has been widely overlooked, and this approach is novel to the best of my knowledge.

[1] Olshausen, B. A. and Field, D. J. (1996). Natural image statistics and efficient coding. *Network*, 7(2):333–339; and Cadieu, C. F. and Olshausen, B. A. (2008). Learning Transformational Invariants from Natural Movies. In *Advances in Neural Information Processing Systems 21*, pages 1–8

[2] Berens, P., Ecker, A. S., Cotton, R. J., Ma, W. J., Bethge, M., and Tolias, A. S. (2012). A fast and simple population code for orientation in primate v1. *The Journal of Neuroscience*, 32(31):10618–10626; and Yaeli, S. and Meir, R. (2010). Error-based analysis of optimal tuning functions explains phenomena observed in sensory neurons. *Frontiers in computational neuroscience*, 4(October):16

### *5.1 Filtering through Point Processes and Optimal Codes*

Though I have introduced the general picture in chapter 2, I will shortly contextualise the framework again. I am interested in modelling a cortical area which is observing spike trains from a population of upstream neurons $N^i(t)$, whose rates depend on a stochastic process $X(t)$ through tuning functions $\lambda^i(x)$. The objective is to estimate the state $X(t)$ from the observations $\boldsymbol{N}(t) = \{N^i(t)\}$ as precisely as possible. Furthermore, I am interested in finding the set of tuning functions $\lambda^j(x)$ driving the observation processes that minimise the MMSE of the estimator $\hat{X}(\boldsymbol{N}_{0:t})$. I will denote the parameters of the encoder by $\varphi$.[3] The optimal encoder is the encoder that minimises the MMSE, which is given by the set of parameters $\varphi^*$, such that

[3] In the case of a population of Gauss-Poisson neurons, the parameters are given by $\varphi = \{\{\theta_i\}, E, \phi\}$.

$$\begin{aligned}\varphi^* &= \operatorname{argmin}_\varphi \boldsymbol{E}\left[\operatorname{Tr}\left[\left(X(t)-\hat{X}(\boldsymbol{N}_{0:t})\right)\left(X(t)-\hat{X}(\boldsymbol{N}_{0:t})\right)^\top\right]\middle| X, \boldsymbol{N}, \varphi\right] \\ &= \operatorname{argmin}_\varphi \operatorname{Tr}\left[\epsilon(\varphi)\right].\end{aligned}$$

Note that differently from chapter 3 I am considering the trace of the MMSE matrix here. This is done to obtain a scalar measure of the quality of a multidimensional estimation problem. The trace gives the sum of the eigenvalues of the MMSE matrix, providing a practical measure of how far from the true value the estimator is on average. In the second line I have also dropped the time dependence as I will be mostly considering stationary results for the MMSE.

#### *Optimal Codes for Control*

In chapter 4 I have introduced the formalism of stochastic optimal control and shown how to extend it to deal with point

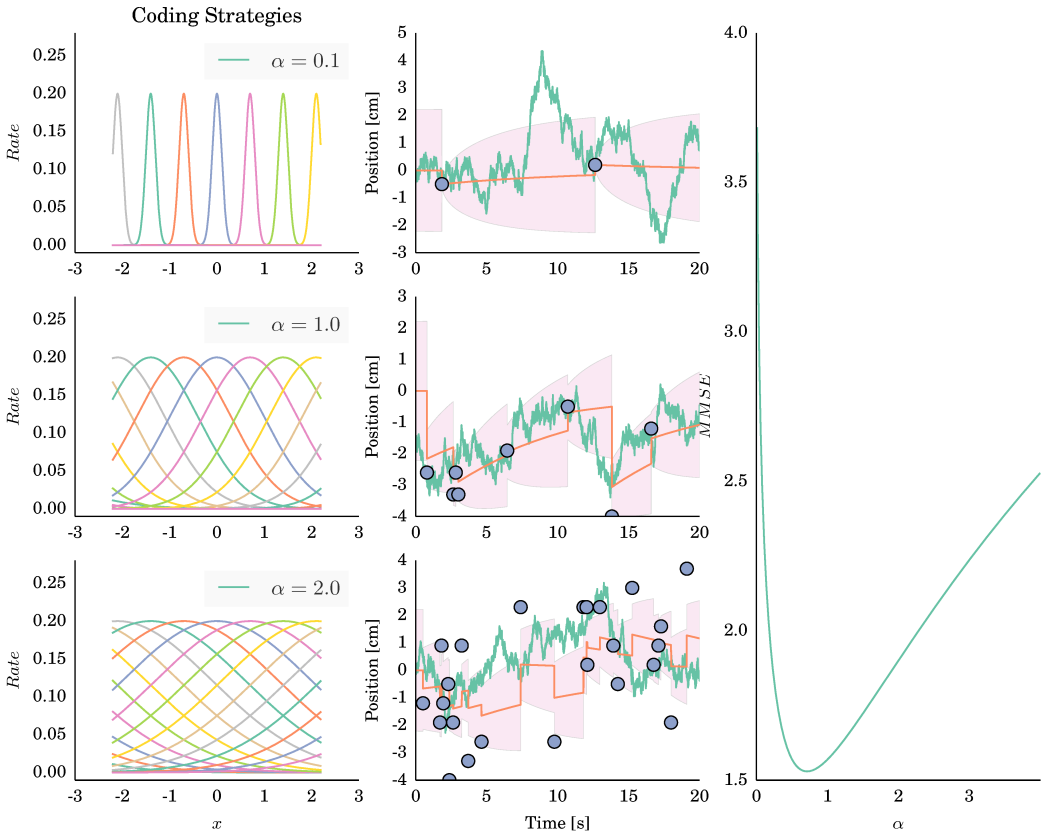


Figure 5.1: Optimal coding for filtering Problems: The leftmost column shows the tuning curves of the neurons in the population. Meanwhile, the middle column shows the general setup of the filtering scheme for each population for the same stochastic process. Note the different situations for narrow and broad tuning curves. The rightmost plot shows the MMSE as a function of the tuning width $\alpha$.

process observations. In a similar way as one can define an optimal encoder for a filtering problem, one can define an optimal encoder for a control problem. Given a control problem with a cost function

$$C(\boldsymbol{N}_{[0,T]}, U_{[0,T]}) = \boldsymbol{E}\left[\int_0^T c(X(t), U(\boldsymbol{N}_{[0,t]}))dt \mid \boldsymbol{N}\right],$$

one can define the optimal encoder to be the one with parameters $\varphi^*$ given by

$$\varphi^* = \text{argmin}_\varphi\, C(\boldsymbol{N}_{[0,T]}, U_{[0,T]}; \varphi).$$

This can lead to different results than the filtering framework as I will show below.[4]

[4] Susemihl, A., Meir, R., and Opper, M. (2014). Optimal Population Codes for Control and Estimation. *ArXiv e-prints*

*Dense Gauss-Poisson Populations*

In this chapter I will mostly discuss results regarding dense populations of Gauss-Poisson neurons. So, unless otherwise noted, I am discussing a population of Poisson neurons with tuning functions $\lambda^m(x)$ given by

$$\lambda^m(x) = \phi \exp\left[-\frac{1}{2}(x-\theta_m)^\top E^\dagger (x-\theta_m)\right]. \tag{5.1}$$

When the stimulus is one-dimensional, I will denote the covariance of the tuning function by $\alpha$ instead of $E$.

The dense coding property holds if the overall firing rate of the population $\hat{\lambda} = \sum_i \lambda^i(x)$ is independent of the stimulus $x$. I will refer to a population of Poisson neurons with Gauss tuning functions such that the dense coding property holds as a dense population of Gauss-Poisson neurons.

## 5.2 *Filtering Linear Stochastic Processes through dense Gauss-Poisson Spike Trains*

Consider a linear stochastic process of the type

$$dX(t) = AX(t) + H^{1/2}dW(t).$$

Though this may seem as a somewhat restrictive choice, a number of processes can be cast into this format. The simple Ornstein-Uhlenbeck process, which I considered in previous chapters is one example, but generalisations to higher dimensions are relatively simple and include, for example, the stochastic damped oscillator. The matrices

$$A = \begin{pmatrix} 0 & 1 \\ -\omega^2 & -2\gamma \end{pmatrix}, \text{ and } H = \begin{pmatrix} 0 & 0 \\ 0 & \eta \end{pmatrix},$$

will lead to a stochastic process with a periodic component, more precisely the stochastic damped oscillator given by the system of SDE's

$$\dot{X}(t) = V(t), \quad dV(t) = -2\gamma V(t)dt - \omega^2 X(t)dt + \eta dW(t).$$

I have here written a pre factor of 2 to the damping coefficient, so that the choice $\gamma = \omega$ leads to the critically damped stochastic oscillator. This is an example of embedding a non-Markov one-dimensional process in a higher-dimensional space to recover the Markov property, as described in section 2.1. In figure 5.2 a few examples of linear stochastic systems are shown, with a couple of random samples of each per plot. Although the focus here is on stationary stochastic processes, this is by no means a necessity for the analysis at hand. Even for non-stationary processes such as the Wiener process, the posterior density can be stationary, allowing us to evaluate the stationary MMSE.

The Wiener process $W(t)$, for example, has a covariance that increases linearly with time.

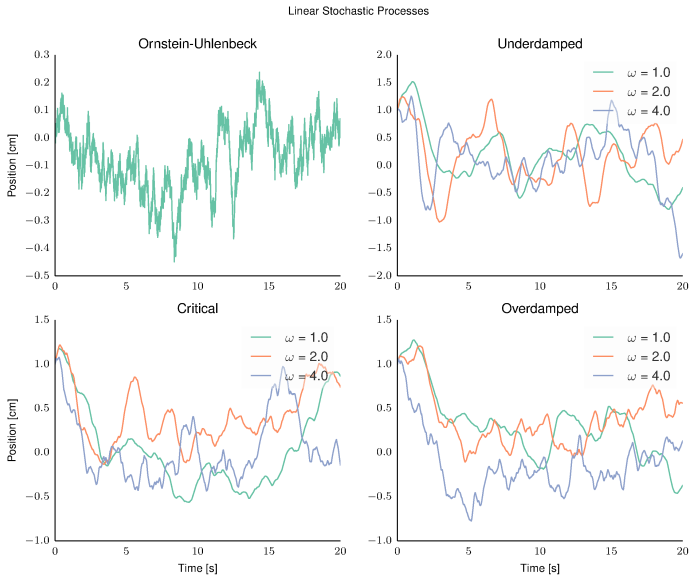


Figure 5.2: Linear stochastic processes: From the top left, we have the one-dimensional Ornstein-Uhlenbeck process, an underdamped stochastic oscillator, a critically damped stochastic oscillator (bottom left), and an over damped stochastic oscillator.

The MMSE $\epsilon(t)$ can be obtained from the formalism derived in chapter 3. Throughout this section I will use both the numerical solution of the evolution equations for $\epsilon(t)$ as well as the mean-field approximation to it.

Let me start with the simplest stationary stochastic process, the Ornstein-Uhlenbeck process given by

$$dX(t) = -\gamma X(t)dt + \eta^{1/2} dW(t),$$

where the exact evolution of the MMSE is given by

$$\frac{d\epsilon(t)}{dt} = -2\gamma\epsilon(t) + \eta - \hat{\lambda}\boldsymbol{E}\left[\frac{s^2}{\alpha^2 + s}\right].$$

I will not consider the temporal evolution of the MMSE, instead I will focus on the stationary value of the MMSE, and therefore on the long-term performance of the encoder in the filtering problem, rather than focusing on the transient, short-time behaviour. I am mostly interested in the dependence of the optimal encoder in the statistical structure of the stimulus, i.e. the correlation timescales and the noise levels. In a natural environment, these changes should be generally slower than the adaptation processes of the sensory apparatus. This is my main motivation to focus on the stationary regime.

In figure 5.3 the equilibrium MMSE of a dense Gauss-Poisson population of neurons encoding an OU process is shown. The dependence on both the maximal firing rate $\phi$ and the tuning width $\alpha$ is shown.

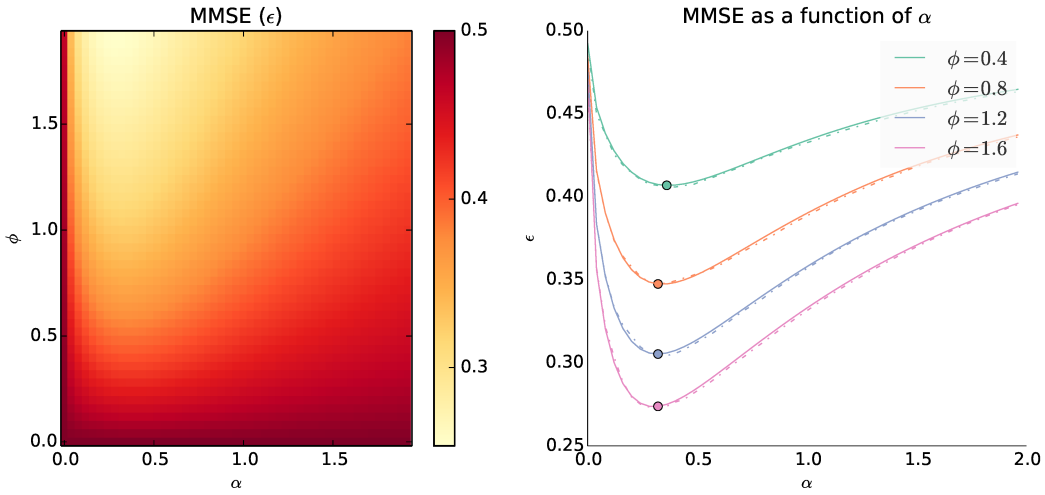


Figure 5.3: Comparing Encoders for Filtering: The left hand panel shows a heat map of the MMSE as a function of the maximal firing rate $\phi$ and the tuning width $\alpha$. There is a trade-off between the number of spikes and the precision of the spikes, manifesting itself in a finite tuning width that minimises the the MMSE. For any $\alpha$ increasing the firing rate $\phi$ simply decreases the MMSE. The right panel shows the dependence in $\alpha$ for a few values of $\phi$.

More interestingly, one can now ask how the optimal encoder depends on any of the parameters of the problem, such as the parameter $\gamma$, for example, which defines the time-scale of correlations in the OU process.[5] In figure 5.4 I have plotted the optimal encoding width $\alpha^*$ as a function of $\gamma$, $\eta$ and $\phi$. It is interesting to be able to provide an accurate account of the dependence of the optimal encoder on the statistical structure of the environment. The leftmost panel shows the dependence of the optimal tuning width in $\gamma$. Shorter time-scales (larger values of $\gamma$) require a higher frequency of spikes, as the information conveyed by those spikes becomes irrelevant more quickly. Thus, holding the maximal firing rate $\phi$ fixed, the only way to increase the frequency of spikes is to have broader tuning functions. Therefore, as $\gamma$ increases, so does $\alpha^*$. Likewise, a higher noise rate $\eta$ leads to the need for more spikes to characterise the system's state, leading to higher values of $\alpha^*$. Increasing the maximal firing rate $\phi$ of the neurons, on the other hand, leads to smaller optimal tuning widths $\alpha^*$.

[5] Remember that the prior kernel of the OU process is given by $k(s,t) = \frac{\eta}{2\gamma}\exp(-\frac{|s-t|}{2\gamma})$.

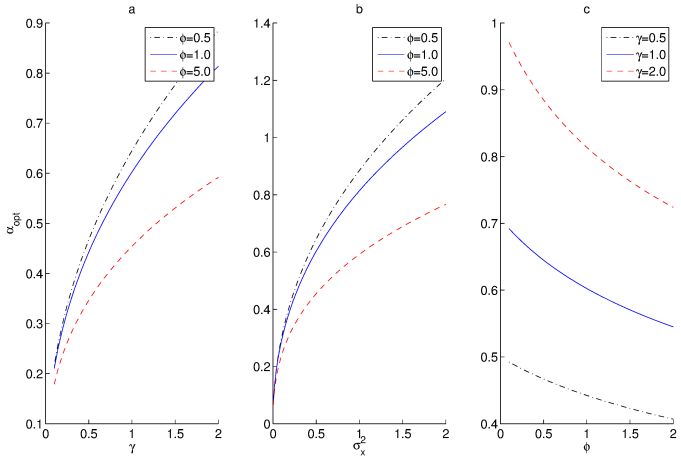


Figure 5.4: Ecological Dependence of the Optimal tuning width for a dense population of Gauss-Poisson neurons encoding the state of an OU process. Increasing the timescale of the correlations $1/2\gamma$ leads to smaller optimal tuning widths, this can be seen in (a). (b) An increase in the noise rate of the process leads to larger optimal tuning widths, as an increase in the variance of the process requires more observations to characterise it. (c) The maximal firing rate of each neurons $\phi$ sets the tradeoff between frequency and precision of the observations. A higher firing rate, tilts the tradeoff towards more precise observations, leading to smaller optimal tuning widths.

*Stochastic Harmonic Oscillator*

A natural extension to consider is the same setup but with the stimulus given by the stochastic harmonic oscillator presented above. Here I will consider a population of neurons whose firing rate depends only on the position of the oscillator, not on the velocity. This would be equivalent to taking the tuning matrix

$$E = \begin{pmatrix} \alpha^2 & 0 \\ 0 & 0 \end{pmatrix},$$

leading to the same form of the tuning functions as before.

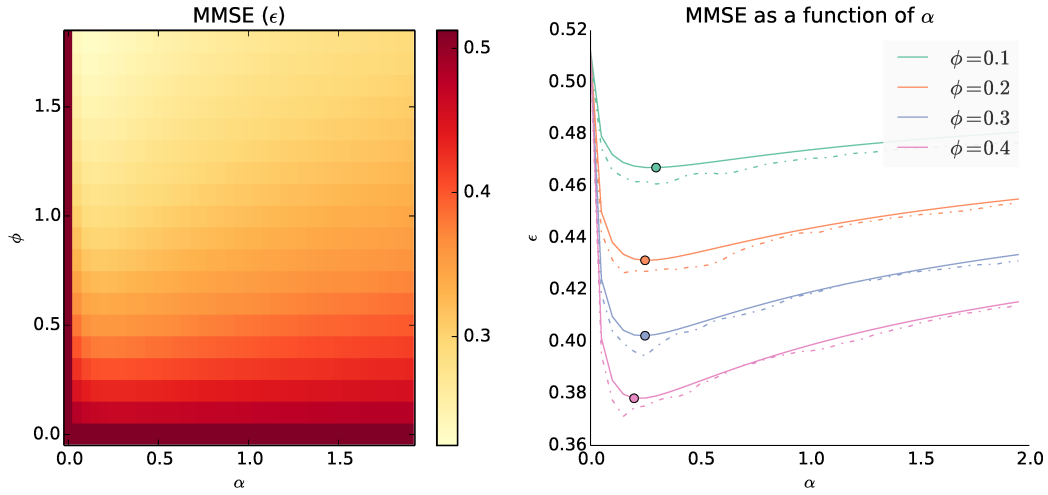


Figure 5.5: Comparing Encoders for Filtering: The left hand panel shows a heat map of the MMSE as a function of the maximal firing rate $\phi$ and the tuning width $\alpha$. There is a trade-off between the number of spikes and the precision of the spikes, manifesting itself in a finite tuning width that minimises the the MMSE. For any $\alpha$ increasing the firing rate $\phi$ simply decreases the MMSE. The right panel shows the dependence in $\alpha$ for a few values of $\phi$.

The results are very similar to the OU case, regardless of the smoother nature of the process considered. In figure 5.5 the MMSE for a dense Gauss-Poisson population coding for a stochastic oscillator is shown. Likewise, figure 5.6 presents the dependence of the optimal tuning width on the parame-

ters of the encoder and the environment. There are three parameters determining the dynamics of the environment, the frequency $\omega$, the damping $\gamma$ and the noise rate $\eta$, and I have presented the dependence of the optimal tuning width on all three. Interestingly, in figure 5.6 (c), one can note a quite different behaviour for the stochastic oscillator. The case $\gamma = 0.5$ represents an underdamped oscillator, $\gamma = 1$ is the critically damped oscillator and $\gamma = 5.0$ an over damped oscillator. First one can note that the dependence of the optimal tuning width on the firing rate $\phi$ is much less pronounced than in the OU process. This is to be expected, as the stochastic oscillator has a smoother structure, and allows one to predict it from past observations more reliably. It can also be noted that the effect of increasing the firing rate on the optimal tuning width is strongest in the underdamped regime, which can also be understood by looking at figure 5.2. When increasing the damping coefficient $\gamma$, the stochastic variations in the velocity become smaller, and the system's state $X(t)$ has smaller, shorter time-scale variations around $X(t) = 0$, while the overall variance of the process also becomes smaller.

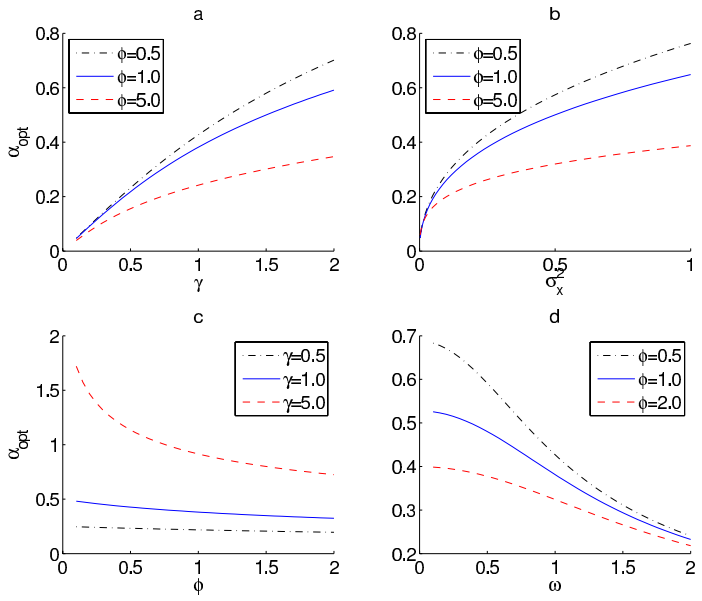


Figure 5.6: Ecological Dependence of the Optimal tuning width for a dense population of Gauss-Poisson neurons encoding the position of a stochastic oscillator. The effect of the damping is similar to the effect of the parameter $\gamma$ in the OU case, as it sets the time-scale of fluctuations in the velocity, which in turn drives the position being estimated. Lower $\gamma$ lead to longer time correlations in the velocity and also in the position, leading to easier reconstruction and a narrower optimal tuning width. The noise intensity $\eta$ also has the same effect as in the OU process, as a stronger noise leads to wider optimal tuning widths. The maximal firing rate of each neuron $\phi$ has a much less pronounced effect on the optimal tuning width than in the OU case, specially for the critical and overdamped regime. This is due to the smoother nature of these processes. The frequency of the process, surprisingly, has the inverse effect as the damping, with higher frequencies leading to lower optimal tuning widths. This is due to the damping effect it has on the velocity, leading to a shorter integration time for the noise in the velocity.

*Smooth Processes*

Through the kernel process formulation developed in section 3.3 one can also treat general Gaussian processes, even non-Markov ones such as the RBF process. By non-Markov processes, I mean processes which can not be rendered Markovian by the inclusion of its derivatives in a higher-dimensional embedding. The MMSE is given by the average posterior variance

of the Gaussian process regression, $\boldsymbol{E}_{\{t_i\}}\left[\Xi(t,t;\{t_i\})\right]$, which can be approximated by the mean-field posterior kernel $g(0)$.[6] In figure 5.7 I have evaluated the kernel mean-field approximation for the RBF kernel $k(s,t) = \exp\left(-(s-t)^2/L\right)$, with $L = 0.5$. Note that the general conclusions drawn in the two previous cases hold here as well, regardless of the more complex temporal structure of the process.

[6] See section 3.3 and appendix A.3.

In the case of linear stochastic processes, I had found that the temporal mean-field approximation of equation (3.7) was surprisingly good at describing the average behaviour of the posterior variance. Therefore it is interesting to study how the mean-field approach performs in the RBF case, where the filtering error is derived from an approximation to the posterior kernel. To that intent, I can evaluate the average posterior variance from section 3.3 numerically. This can be done by generating a large number of Poisson spike trains and evaluating the posterior covariance $\Xi(t;\{t_i\})$ for each and taking an average. This is also shown in figure 5.7 as the dotted lines. The averaging in this case is much more costly, as for every set of $M$ spike times, one needs to invert an $M \times M$ matrix, which takes of the order of $M^3$ operations. As can be seen from the figure, the mean-field approximation still agrees very well with the numerical average, leading to undistinguishable optimal tuning widths.

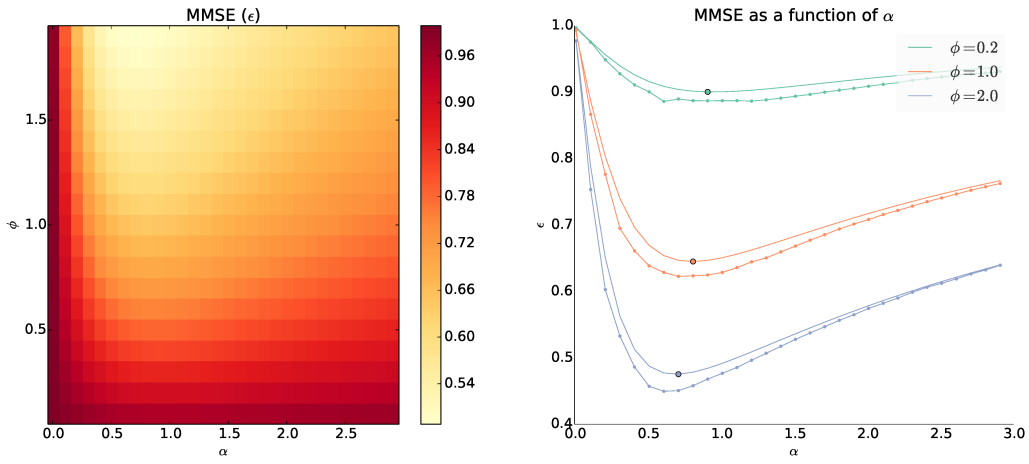


Figure 5.7: MMSE for a dense population of Gauss-Poisson neurons encoding a RBF process. The MMSE follows the same trends as for the OU process and the stochastic harmonic oscillator. The dotted lines show the numerically averaged MMSE, obtained directly from the Gaussian process regression.

## 5.3 *Moving Away From Gaussian Distributions*

In chapter 2 I have presented filtering tools for general stochastic processes and observation processes, namely the ADF and particle filter techniques. So far, in the analysis of optimal codes I have restricted myself to the dense coding limit, which significantly simplifies the analysis, rendering the Gaussian

ADF approach exact. What happens when the posterior is not Gaussian, though? I will consider a couple of interesting cases shortly. There are two ways one can leave the dense coding limit. The first one is to have a population which does not densely cover the stimulus space. The simplest case would be a single neuron with a Gaussian tuning function,[7] for example, which could clearly not cover the stimulus space. The second possibility refers to the nature of the neurons. If their spiking is time-dependent or adaptive, the homogeneity of the firing rate will break and we will have a stimulus-dependent population firing rate as well. Of course the posterior can also be non-Gaussian due to the prior. That would mean the process being observed is non-Gaussian. I will consider the simple case of a stochastic process in a double-well potential, as an example of non-Gaussian processes.

[7] See figure 3.1.

Once again my main interest is in the optimality of said codes. It is straightforward to estimate the MMSE of the mentioned cases from simulations directly. Though much less practical than the dense coding case, where one could refrain from simulating the stimulus trajectories and spike trains, the principles remain the same. It is not true, however, that the average posterior covariance gives the MSE of the estimator in this case. Remember that I have used that the estimator is the posterior average to show that, and this does not hold generally for either the ADF or the particle filter estimator. For the particle filter, it can be shown that in the limit of many particles the empirical distribution converges to the posterior distribution, yielding the posterior mean estimator in the limit of infinite particles.[8] The ADF estimator, however, has no simple relation to the posterior mean estimator, and further has no guarantee of converging to the true posterior. So in both cases we are forced to work directly with the average estimation error to obtain a measure of the MSE.

[8] Crisan, D. and Doucet, A. (2002). A survey of convergence results on particle filtering methods for practitioners. *Signal Processing, IEEE Transactions on*, 50(3):736–746

This section is meant to illustrate the application of the framework of MSE-optimal codes outside of the assumptions made in the previous section. Though optimising a code for the MSE is not as straightforward for more complex, higher-dimensional problems, this shows that it is in principle possible, and the hurdles are mostly of implementation, rather than conceptual.

*Sparse Populations*

The simplest form of breaking the dense coding assumption is, well, not having dense populations. The most extreme case would be if there were only one or a few neurons coding for the stimulus. In that case, the population firing rate could be very strongly dependent on the state of the system. I will consider a simple case here to illustrate the applicability of

the MSE method. I will consider the case of a population of two neurons, equidistant from the mode of the stimulus distribution, and investigate the dependence of the MSE on the separation between the two and the width of the tuning functions.

I will use three filtering schemes for this problem and compare them. First, I consider the filtering equations given by equation (2.21). In the dense coding case, these equations are exact, but one can use them as an approximate filtering method regardless of the population being dense. This amounts to ignoring the probability of a spike not being fired, and looking only at the probability of a spike being fired when a spike is actually observed. I will refer to this filtering scheme as the Dense ADF approach. A second possibility is to apply the ADF approach with a Gaussian distribution, now taking the rate terms in equation (2.27) into account and updating the mean and covariance through equation (2.20) upon the observation of a spike. I will refer to this option as the Full ADF approach. A third possibility is to use a particle filter with a large number of particle (I have taken $M = 1000$ here) to estimate the posterior probability given the spikes. This again, takes into account the probability of a spike not having been fired in every time instant by every neuron.

Assuming both neurons have Gaussian tuning functions with the same tuning width, one could ask what is the best tuning width and spacing between the two neurons to optimise the MSE. In figure 5.8 I have plotted the MSE as a function of $\Delta\theta$ and $\alpha$, showing a clear optimum. There are some artefacts due to the interplay of $\alpha$ and $\Delta\theta$, but all three approaches show a clear minimum. Surprisingly, even in this case, the dense ADF approach yields nearly the same MSE as the particle filter or the full ADF approach, showing a relative absolute deviation from the full ADF approach of less than 1% and of approximately 1.5% from the particle filter approach. This might be due to the Gaussian nature of the stimulus, but it is an interesting point for further investigation.

*Adaptive Neurons*

A different situation which could lead to non-uniform firing rates is adaptation in the firing rate of the neurons, even in a dense population of neurons. The spike-frequency adaptation process described in section 2.5 is a simple model where the essence of neural adaptation is already present. I will consider the impact of adaptation on the MMSE of a simple linear stochastic process as we have considered above.

A number of interesting questions arise with respect to adaptation in neurons. The adaptation implemented by the spike rate modulation $\kappa(t)$ is similar to the spike-frequency adap-

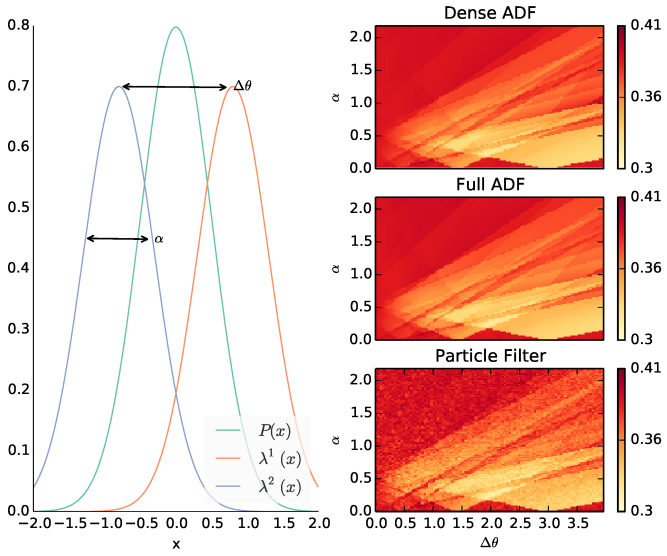


Figure 5.8: Two neurons with Gaussian tuning functions coding for a simple OU process. The left panel shows two sample tuning functions, explaining the parameters $\Delta\theta$ and $\alpha$. The left panel shows the MSE according to the different approaches. The top shows the ADF approach assuming a dense population (i.e. ignoring the firing rates in the absence of spikes). The middle plot shows the MSE obtained with a full Gaussian ADF approach. The bottom plot shows the MSE of a particle filter. Though the three approaches give different results, the general dependence of the encoder on the parameters is clear for all three approaches. There is an optimal value of $\Delta\theta$ and $\alpha$ which minimises the MSE in all three approaches

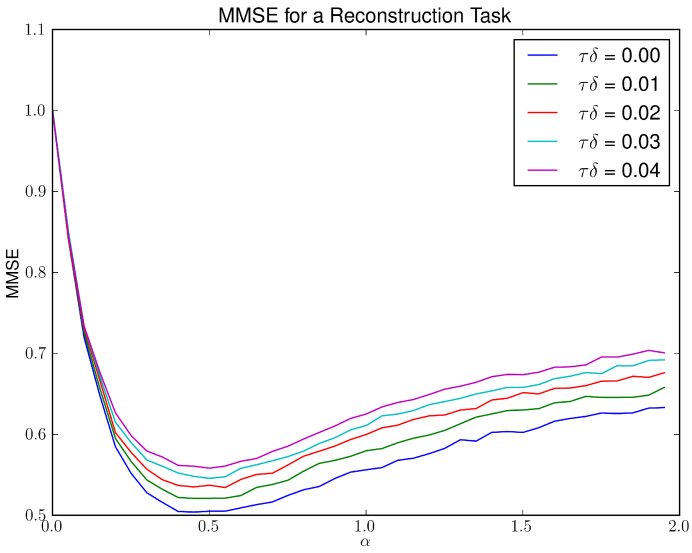


Figure 5.9: The MSE of a reconstruction task from the observation of adaptive spike trains. The product $\tau\delta$ quantifies the intensity of the adaptation. As in the previous cases, one can note the existence of a finite optimal tuning width.

tation often described in neuroscience.[9] As can be seen in figure 5.9, at first glance it would seem that adaptation does not help the population code. This is somewhat misleading, though, as the stronger the adaptation time-scale $\tau$, the lower the firing rate of the population. It would make sense, then to compare adaptive and non-adaptive populations with the same firing rate, to account for the effect of the adaptation. This is shown in figure 5.10. Here the mean-squared error of the filter is plotted as a function of the firing rate of the population, and it is immediately obvious that the adaptive populations achieve a better performance with a much lower firing rate.

[9] Benda, J. and Herz, A. V. (2003). A universal model for spike-frequency adaptation. *Neural computation*, 15(11):2523–2564

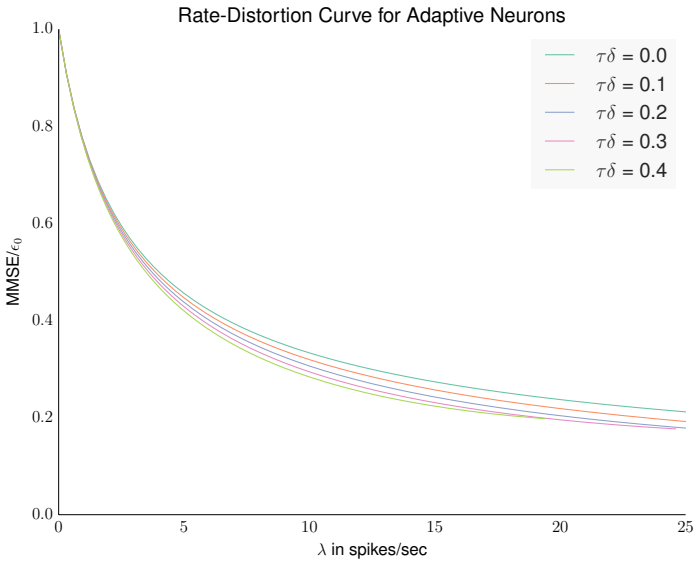


Figure 5.10: MMSE of adaptive population as a function of the population firing rate. Though the adaptation leads to a worse reconstruction error when keeping all other parameters fixed, here one can see that it allows for a more precise reconstruction with the same amount of spikes. There is a saturation in the improvement however, as the adaptation puts a limit on the number of spikes the population can fire.

*Nonlinear Stochastic Processes*

A different source of non-Gaussian distributions can be found directly in the stochastic process one is modelling as stimuli. A simple model that leads to non-Gaussian distribution is a stochastic process in a double well potentialo, given by $V(x) = \nu(x+x_0)^2(x-x_0)^2$. This potential will have two stable points at $x = \pm x_0$. One can then define a stochastic system moving in this potential as

$$dX(t) = -\frac{dV}{dx}dt + \eta^{1/2}dW(t).$$

This leads to

$$dX(t) = 4\nu X(t)\left(x_0 - X(t)^2\right)dt + \eta^{1/2}dW(t).$$

Figure 5.11 shows some sample paths from this process.

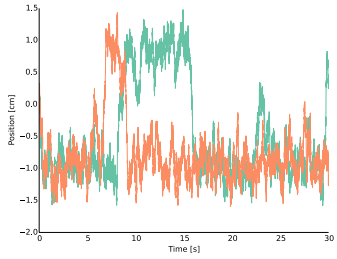


Figure 5.11: Samples of the Bistable process mentioned in the text.

Following the framework of chapter 2, it is simple to devise a particle filtering algorithm for this system. Assuming the observations are still from a dense code, one can simply take the particles $Z^i(t)$ evolving by the same SDE as the system. So after discretising one will have

$$\Delta Z^i(t) = 4\nu Z^i(t)\left(x_0 - Z^i(t)^2\right)\Delta t + \sqrt{\Delta t \eta}\, V^i(t),$$

where each of the $V^i(t)$ is a standard normal random variable independent of all other $V^j(t)$. The weights $w^i(t)$ associated with each particle will then be updated as

$$w^i(t) = w^i(t - \Delta t)\lambda^j\left(Z^i(t)\right),$$

in case neuron $j$ spikes and simply left unchanged in the absence of spikes. Here $\lambda^j(z)$ are the usual Gaussian tuning functions given by equation (5.1).

Alternatively one can develop an ADF algorithm for the proposed system. Using the relationships derived in chapter 3 it is easy to obtain the equations for $\mu(t)$ and $\Sigma(t)$,

$$\frac{d\mu(t)}{dt} = 4\nu\mu(t)\left(x_0 - \mu(t)^2 - 3\Sigma(t)\right), \tag{5.2a}$$

and

$$\frac{d\Sigma(t)}{dt} = 8\nu\Sigma(t) - 24\mu(t)^2\Sigma(t) - 24\Sigma(t)^2 + \eta. \tag{5.2b}$$

In figure 5.12 I show both approaches applied to the bistable problem. We can then leverage the particle filtering approach, which shows better results for this filtering problem and look at the optimal tuning width for an estimation task. The MMSE for the bistable process is shown in figure 5.13. The general conclusions arrived at previously hold in this setting as well and the MSE is minimal for a finite tuning width, underlining the trade-off between precision and frequency discussed above.

## *5.4 Optimal Codes for Control*

I have extensively argued for the usefulness of accurate estimation of a system's state when interacting with it. Though this is by no mean false, real world systems are often very high-dimensional, or at least represented in a very high-dimensional code, leading to trade-offs when deciding where to allocate sensory resources. One simple example is the density of photoreceptors in the retina. If we assume the retina evolved to allow for optimal estimation of the visual scene facing an animal, we would expect it to cover the entire visual field evenly. That is of course not the case, and there are a number of anomalies in the distribution of receptive fields which

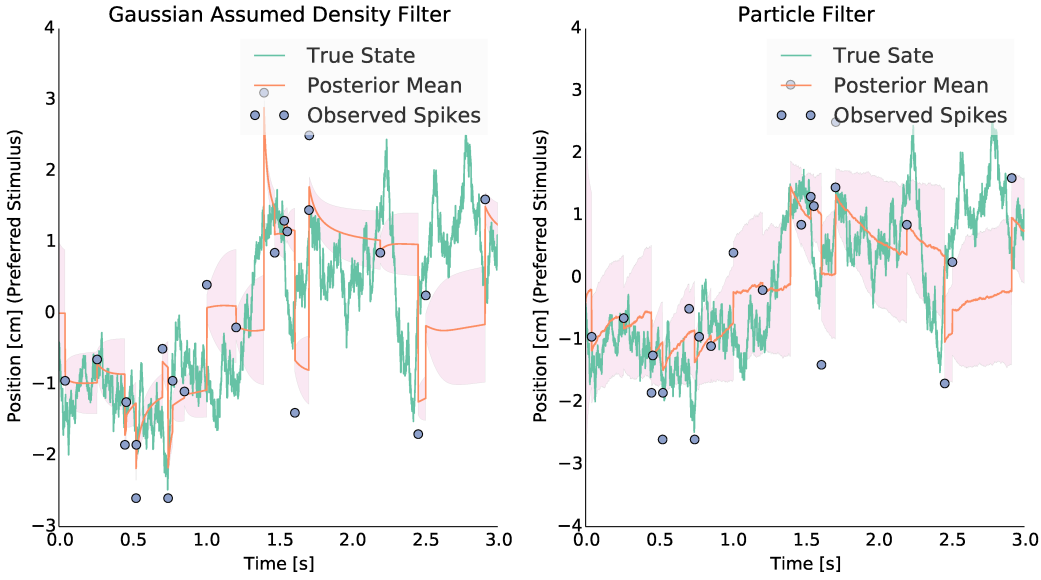


Figure 5.12: ADF (left) and particle (right) filters applied to the dense Gauss-Poisson coding of a bistable process. Note that the ADF filter consistently underestimates the variance of the posterior distribution. Leading it to lose track of the system's state at some points.

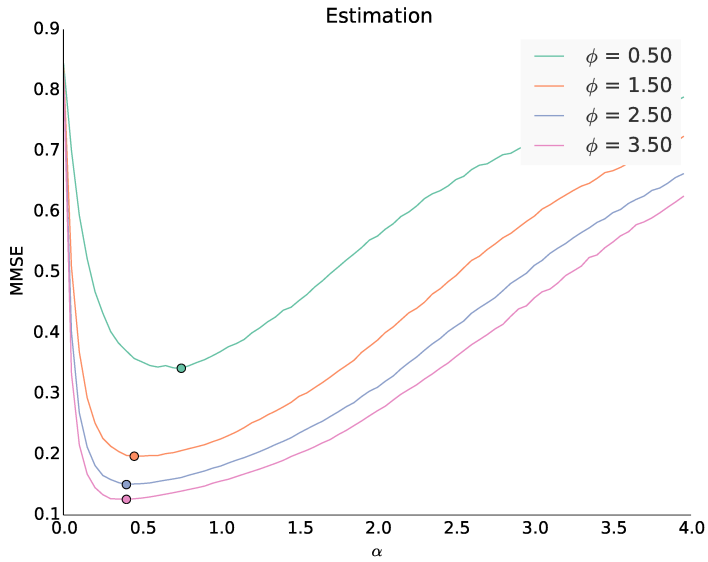


Figure 5.13: Dependence of the MSE on the tuning width $\alpha$. The width follows the same trend as for the previously considered processes. The tuning width decreases with increasing firing rates. However, here one can see, that the minimum is less sharp, due to the bistable nature of the process. If a spike allows one to discern between the two stable states, it already contributes a lot to the estimation of the state $X(t)$.

can be attributed to the importance of different visual queues for decision making and risk assessing. A simple example of which is the distribution of photoreceptors in the retina, with the concentration of cones varying up to two orders of magnitude between the periphery and the fovea.[10]

What could be the reasons for an optimal code for an estimation problem to be sub-optimal for a control problem? I will present examples that show two possible reasons for different optimal coding strategies in estimation and control. First, one should note that control problems are often defined over a finite time horizon. One set of classical experiments involves reaching for a target under time constraints.[11] If one takes the maximal firing rate of the neurons ($\phi$) to be constant while varying the width of the tuning functions, this will lead the number of observed spikes to be inversely proportional to the precision of those spikes, forcing a trade-off between the number of observations and their quality. This trade-off can be tilted to either side in the case of control depending on the information available at the start of the problem. If one is given complete information on the system state at the initial time 0, the encoder needs fewer spikes to reliably estimate the system's state throughout the duration of the control experiment, and the optimal encoder will be tilted towards a lower number of spikes with higher precision. Conversely, if at the beginning of the experiment one has very little information about the system's state, the encoder will be forced towards lower precision spikes with higher frequency.

Secondly, one should note that the optimal encoder for estimation does not take into account the differential weighting of different dimensions of the system's state. When considering a multidimensional estimation problem, the optimal encoder will generally allocate all its resources equally between the dimensions of the system's state. In the framework presented below one can think of the dimensions as the singular vectors of the tuning matrix $E$ and the resources allocated to it as the singular values. In this sense, I will consider a set of coding strategies defined by matrices $E$ of constant determinant. This constrains the overall firing rate of the population of neurons to be constant, and one can then consider how the population should best allocate its observations between these dimensions. Clearly, in an anisotropic control problem, which places a higher importance in controlling one dimension, the optimal encoder for the control problem will be expected to allocate more resources to that dimension. This is indeed shown to be the case for the Poisson codes considered, as well as for a simple LQG problem with Gaussian observations in continuous time when we constrain the noise covariance to have the same structure.

[10] Curcio, C. A., Sloan, K. R., Packer, O., Hendrickson, A. E., and Kalina, R. E. (1987). Distribution of cones in human and monkey retina: individual variability and radial asymmetry. *Science*, 236(4801):579–582

[11] Battaglia, P. W. and Schrater, P. R. (2007). Humans trade off viewing time and movement duration to improve visuomotor accuracy in a fast reaching task. *The Journal of neuroscience*, 27(26):6984–94

*The Trade-off Between Precision and Frequency of Observations*

In this section I consider populations of neurons with tuning functions as given by equation (2.18) having tuning centers $\theta_m$ distributed along a one-dimensional line. In the case of a stimulus modelled by the Ornstein-Uhlenbeck process these will be simply one-dimensional values $\theta_m$ whereas in the case of the stochastic oscillator, I will consider tuning centres of the form $\theta_m = (\eta_m, 0)^\top$, filling only the first dimension of the stimulus space. This means that in the case of the stochastic oscillator, the observer does not have direct access to the velocity of the system, only of its position. Note that in both cases the (dense) population firing rate $\hat{\lambda} = \sum_m \lambda_m(x)$ will be given by $\hat{\lambda} = \sqrt{2\pi}\alpha\phi/|\Delta\theta|$, where $\Delta\theta$ is the separation between neighbouring tuning centres $\theta_m$.

The OU process controlled by a process $U(t)$ is given by the SDE

$$dX(t) = (bU(t) - \gamma X(t))dt + D^{1/2}dW(t),$$

and the control problem is defined by a cost function

$$C(X,U) = \int_0^T \left(X(t)^\top Q X(t) + U(t)^\top R U(t)\right) dt.$$

Equation (4.16) can then be evaluated by simulating the dynamics of $\Sigma(t)$. This is exactly the problem solved in chapter 3 and it has extensively been discussed therein.[12] Following those results one can also approximate the average of the posterior variance by a mean-field formalism which works surprisingly well. The evolution of the average posterior variance is given by the average of equation (2.21b), which involves nonlinear averages over the covariances. The mean-field evolution of $\boldsymbol{E}\left[\Sigma(t)\middle|\Sigma_0\right]$ is given by

[12] See also (Susemihl et al., 2013).

$$\frac{d\boldsymbol{E}\left[\Sigma(t)\right]}{dt} = A\boldsymbol{E}\left[\Sigma(t)\right] + \boldsymbol{E}\left[\Sigma(t)\right]^\top A^\top + D - \hat{\lambda}\boldsymbol{E}\left[\Sigma(t)\right] E^\dagger \boldsymbol{E}\left[\Sigma(t)\right]\left(I + E^\dagger \boldsymbol{E}\left[\Sigma(t)\right]\right)^{-1}.$$

To assess the quality of this approximation I have also computed the averages of $\Sigma(t)$ numerically with a large number of sample paths to compare to the mean-field approximation. In (Susemihl et al., 2011) and (Susemihl et al., 2013) I had reported a very good agreement in the mean-field and numerically calculated values of $\boldsymbol{E}\left[\Sigma(t)\right]$. $f(\Sigma, t)$ however is an integral over time of this average, so one can expect that the deviation between the mean-field approximation and the numerical average will be larger. As can be seen in figure 5.14, the mean-field approximation is not as precise as in the case of the simple average, but the dependence of the average on $\alpha$, however, remains very well explained by the mean-field approach.

Alternatively, one can look at a system with more complex dynamics, and I will take as an example the stochastic damped harmonic oscillator given by the system of equations

$$\dot{X}(t) = V(t), \quad dV(t) = \big(bU(t) - \gamma V(t) - \omega^2 X(t)\big)\,dt + \eta^{1/2} dW(t). \tag{5.3}$$

Furthermore, I assume that the tuning functions only depend on the position of the oscillator, therefore not giving any information about the velocity. The controller in turn seeks to keep the oscillator close to the origin while steering only the velocity. This can be achieved by the choice of matrices

$$A = \begin{pmatrix} 0 & 1 \\ -\omega^2 & -\gamma \end{pmatrix}, B = \begin{pmatrix} 0 & 0 \\ 0 & b \end{pmatrix}, D = \begin{pmatrix} 0 & 0 \\ 0 & \eta^2 \end{pmatrix},$$

$$R = \begin{pmatrix} 0 & 0 \\ 0 & r \end{pmatrix}, Q = \begin{pmatrix} q & 0 \\ 0 & 0 \end{pmatrix} \text{ and } E = \begin{pmatrix} \alpha^2 & 0 \\ 0 & 0 \end{pmatrix}.$$

In chapter 4 I have argued that a good way to quantify the effect of the encoder on the costs of a control problem is the function $f(\Sigma, t)$ derived there. This quantifies the effect of the uncertainty resulting from estimating the system's state through that encoder on the control costs. In figure 5.14 I have plotted the uncertainty-dependent costs $f$ for LQG control, for the Poisson observed control, as well as the MMSE for the Poisson filtering problem and for a Kalman-Bucy filter with a same noise covariance equal to the tuning matrix $E$. This illustrates nicely the difference between Kalman filtering and the Gauss-Poisson filtering considered here. The Kalman filter MSE has a simple, monotonically increasing dependence on the noise covariance, and one should simply strive to design sensors with the highest possible precision ($\alpha = 0$) to minimise the MMSE and control costs. The Poisson case leads to optimal performance at a non-zero value of $\alpha$. Importantly the optimal values of $\alpha$ for estimation and control differ. Furthermore, in view of section 3.4, I have also plotted the mutual information between the process $X(t)$ and the observation process $N(t)$, to illustrate that information-based arguments would lead to the same optimal encoder as MMSE-based arguments.

The result that the MMSE-optimal encoder also maximises the mutual information had not been previously reported, and is most likely a consequence of the Gaussian nature of the distributions considered. It can be shown to hold exactly in the mean-field approximate, as well. The mutual information quantifies the information contained in the observations about the stimulus. The MSE on the other hand quantifies the second moment of the difference between the estimate and the true value of the stimulus. If all distributions are Gaussian, these two quantities are related as all the information

the observations can contain are condensed into the two first moments of the posterior distribution. This does not need to be the case generally, but provides an interesting point of entry to further research questions.

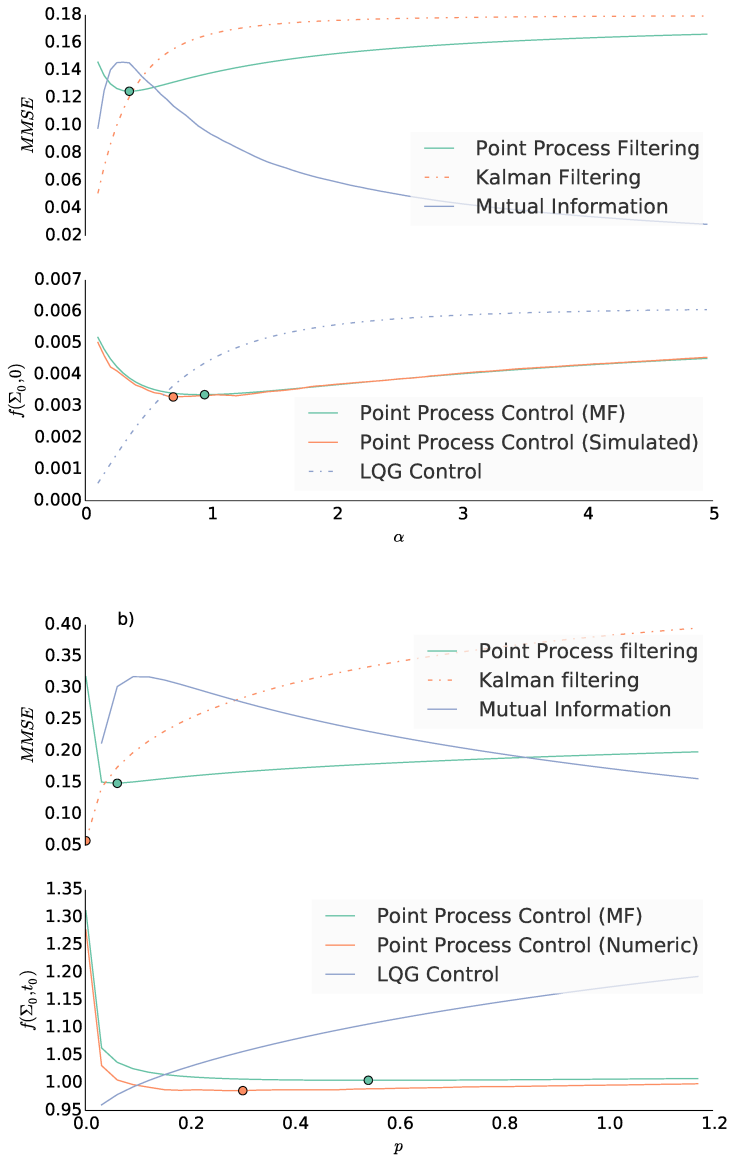


Figure 5.14: The trade-off between the precision and the frequency of spikes is illustrated for the OU process (top) and the stochastic oscillator (bottom). In both figures, the initial condition has a very uncertain estimate of the system's state, biasing the optimal tuning width towards higher values. This forces the encoder to amass the maximum number of observations within the duration of the control experiment. Parameters for figure (a) were: $T = 2, \gamma = 1.0, \eta = 0.6, b = 0.2, \phi = 0.1, \Delta\theta = 0.05, Q = 0.1, Q_T = 0.001, R = 0.1$. Parameters for figure (b) were $T = 5, \gamma = 0.4, \omega = 0.8, \eta = 0.4, r = 0.4, q = 0.4, Q_T = 0, \phi = 0.5, \Delta\theta = 0.1$.

*Allocating Observation Resources in Anisotropic Control Problems*

A second factor that could lead to different optimal encoders in estimation and control is the structure of the cost function $C$. Specifically, if the cost functions depends more strongly on a certain coordinate of the system's state, uncertainty in that particular coordinate will have a higher impact on expected future costs than uncertainty in other coordinates. I

will here consider two simple linear control systems observed by a population of neurons restricted to a certain firing rate. This can be thought of as a metabolic constraint, since the regeneration of membrane potential necessary for action potential generation is one of the most significant metabolic expenditures for neurons.[13] This will lead to a trade-off, where an increase in precision in one coordinate will result in a decrease in precision in the other coordinate.

[13] Attwell, D. and Laughlin, S. B. (2001). An energy budget for signaling in the grey matter of the brain. *Journal of Cerebral Blood Flow & Metabolism*, 21(10):1133–1145

I consider a population of neurons whose tuning functions cover a two-dimensional space. Taking a two-dimensional isotropic OU system with state $X(t) = (X_{1,t}, X_{2,t})^\top$ where both dimensions are uncoupled, one can consider a population with tuning centres $\theta_m = (\eta_1^m, \eta_2^m)^\top$ densely covering the stimulus space. To consider a smoother class of stochastic systems I will also consider a two-dimensional stochastic oscillator with state $X(t) = (X_1(t), V_1(t), X_2(t), V_2(t))^\top$, where again, both dimensions are uncoupled, and the tuning centres of the form $\theta_m = (\eta_1^m, 0, \eta_2^m, 0)^\top$, covering densely the position space, but not the velocity space.

Since I am interested in the case of limited resources, I will restrict myself to populations with a tuning matrix $E$ yielding a constant population firing rate. One can parametrise these simply as

$$E_{OU}(\zeta) = p^2 \begin{pmatrix} \tan(\zeta) & 0 \\ 0 & \operatorname{cotan}(\zeta) \end{pmatrix}$$

for the OU case and

$$E_{Osc}(\zeta) = p^2 \begin{pmatrix} \tan(\zeta) & 0 & 0 & \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \operatorname{cotan}(\zeta) & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

for the stochastic oscillator, where $\zeta \in (0, \pi/2)$. This will yield the firing rate $\hat{\lambda} = 2\pi p^2 \phi / (\Delta\theta)^2$, independent of the angle $\zeta$.

One can then compare the performance of all observers with the same firing rate in both control and estimation tasks. As mentioned, I am interested in control problems where the cost functions are anisotropic, that is, one dimension of the system's state vector contributes more heavily to the cost function. To study this case I will consider cost functions of the type

$$c(X(t), U(t)) = Q_1 X_1(t)^2 + Q_2 X_2(t)^2 + R_1 U_1(t)^2 + R_2 U_2(t)^2.$$

This again, can be readily cast into the formalism introduced above, with a suitable choice of matrices $Q$ and $R$ for both the OU process as for the stochastic oscillator. I will look at the t

the case where the first dimension of $X(t)$ contributes more strongly to the state costs (i.e., $Q_1 > Q_2$).

The filtering error can be obtained from the formalism developed in chapter 3 in the case of Poisson observations and directly from the Kalman-Bucy equations in the case of Kalman filtering. For LQG control, one can simply solve the control problem for the system mentioned through the Ricatti equation and obtain an estimate of the uncertainty-related costs (see e.g. Åström (2006, p.288)). The Poisson-coded version of the control problem can be solved using either direct simulation of the dynamics of $\Sigma(t)$ or by a mean-field approach which has been shown to yield excellent results for the system at hand. These results are summarised in figure 5.15, with similar notation to that in figure 5.14. Note the extreme example of the stochastic oscillator, where the optimal encoder is concentrating all the resources in one dimension, essentially ignoring the second dimension.

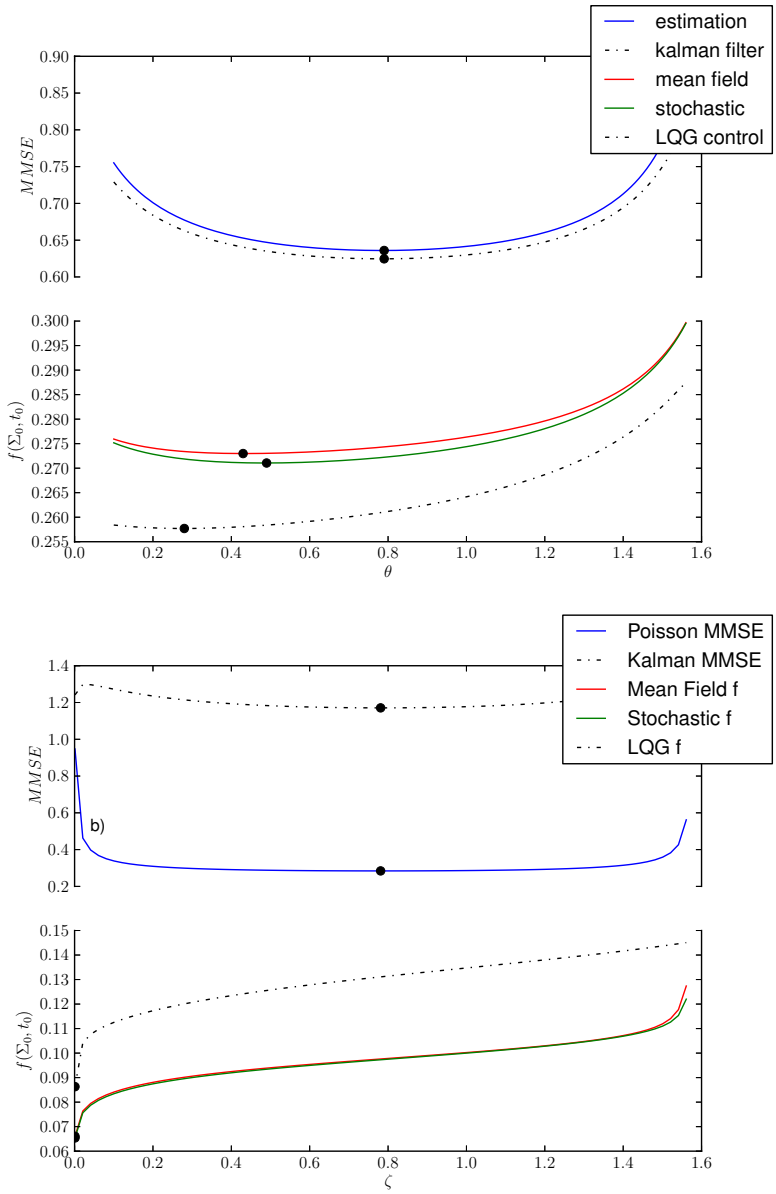


Figure 5.15: The differential allocation of resources in control and estimation for the OU process (top) and the stochastic oscillator (bottom). Even though the estimation MMSE leads to a symmetric optimal encoder both in the Poisson and in the Kalman filtering problem, the optimal encoders for the control problem are asymmetric, allocating more resources to the first coordinate of the stimulus.

# 6
# *Discussion*

I will shortly summarise the main findings discussed in each chapter, and will then discuss the impact and relevance of the work.

*Chapter 2*

In chapter 2 I have reviewed the theory of stochastic filtering, with a focus on doubly stochastic point process observations. I have provided an informal derivation for the Snyder equation which is novel to the best of my knowledge. I have also reformulated the fast population coding approach to the language of stochastic calculus, which proved useful to derive expressions for the posterior covariance.

*Chapter 3*

For the matched case, the MSE of the posterior mean estimator is given by the average covariance matrix of the estimator. Using this, I have formulated the time-dependent MSE as an average over a drift-jump stochastic process and used the language of statistical physics to treat it in the stationary regime. For the Ornstein-Uhlenbeck process, which provides a stereotypical stationary stochastic process, I have shown a closed-form expression for the stationary distribution of covariances. This distribution shows a particular divergence when the interspike interval of the observation process is shorter than the correlation time of the observed process. A similar solution has also been derived for the limit of small firing rates. I have also provided a Van Kampen approximation of the stationary distribution, which allowed me to establish a system size variable for the problem at hand.

Though most of my work on the MSE relied on the assumption of a Markovian stimulus, the analysis can also be extended to general Gaussian processes. This is done by defining a stochastic process taking values in the space of kernels.

In that way, by deriving an approximate solution, one can estimate the MSE of a filtering problem of any type of Gaussian process. This has been illustrated for the RBF kernel, leading to infinitely smooth random processes.

*Chapter 4*

In chapter 4 I have dealt with control theory. Though the study of control theory has been gaining traction continuously in the computational neuroscience community, the issue of optimal coding for control problems had been barely touched. I have presented the formalism of stochastic optimal control, using the Hamilton-Jacobi-Bellman equation, and have then applied this to a belief state formulation of the filtering problem I dealt with in the previous chapters. In the limit of a dense population of Gauss-Poisson neurons, I have been able to derive an exact solution for the optimal cost-to-go making the uncertainty-dependent portion of the costs explicit. This could be solved using a Feynman-Kac approach, and can be evaluated as an average over paths of the covariance process. This is to the best of my knowledge a novel result.

*Chapter 5*

I then turned to applications of my results. With the formalism in hand, I have treated a number of filtering problems. For the dense Gauss-Poisson populations, I have shown results for the OU process, the stochastic harmonic oscillator and the RBF process. I have also shown an approximate treatment of a bistable stochastic process. The general results seem to be very robust to the nature of the process being observed. To illustrate the applicability of the method, I have also treated a number of other cases approximately.

The comparison between control-optimal and estimation-optimal codes is a central finding of this work. Though the situations that lead to it might look trivial at first, there has been little to no attention devoted to the study of optimal codes in a control-theoretical setting. I have argued that this may be due to the fact that when dealing with Gaussian additive noise, the optimal encoder is trivial. This is not the case for Poisson processes, where the rate and precision of observations is coupled.

*Discussion*

I have extensively discussed methods for deriving optimal codes in a population of neurons. This seems a worthwhile research

avenue to me, as it has already bore fruits in the understanding of the nervous system, as well as lead to a number of technical advances in statistical methods. The fields of computer vision, machine learning and robotics stand to profit a lot of findings on optimal codes for distributed systems such as the ones studied here. The tendency in science, as well as in technology, seems to be towards decentralisation and autonomy. So, the study of optimal sensing for distributed systems is of great interest.

I believe the contribution of this thesis to be a small step toward an understanding of the brain in terms of the inferences performed by it. Though I have restricted myself to some specific cases, this was mainly to explore the analytic results obtained, which allowed for deeper insight. As I have shown in chapter 5, more general setups can easily be cast into an optimal coding framework where the MSE is taken as an objective function to be minimised.

Another important issue I have touched upon in this thesis is the consideration of control problems when looking at optimal population codes. Though this is still in an incipient stage, I think the use of control problems to determine optimal coding strategies could yield important insights with regard to adaptation and efficient coding. I have been able to show a simple result for the case of a dense population of neurons, and I believe the development of approximate methods for that would be an interesting research direction to investigate.

# 7
# Bibliography


Ahmadian, Y., Pillow, J. W., and Paninski, L. (2011). Efficient Markov chain Monte Carlo methods for decoding neural spike trains. *Neural Computation*, 23(1):46–96.

Atick, J. J. (1992). Could information theory provide an ecological theory of sensory processing? *Network: Computation in neural systems*, 3(2):213–251.

Attwell, D. and Laughlin, S. B. (2001). An energy budget for signaling in the grey matter of the brain. *Journal of Cerebral Blood Flow & Metabolism*, 21(10):1133–1145.

Battaglia, P. W. and Schrater, P. R. (2007). Humans trade off viewing time and movement duration to improve visuomotor accuracy in a fast reaching task. *The Journal of neuroscience*, 27(26):6984–94.

Bear, M. F., Connors, B. W., and Paradiso, M. A. (2007). *Neuroscience*, volume 2. Lippincott Williams & Wilkins.

Beck, J. M., Latham, P. E., and Pouget, A. (2011). Marginalization in neural circuits with divisive normalization. *The Journal of neuroscience*, 31(43):15310–9.

Bell, A. J. and Sejnowski, T. J. (1997). The independent components of natural scenes are edge filters. *Vision research*, 37(23):3327–3338.

Bellman, R. (1952). On the theory of dynamic programming. *Proceedings of the National Academy of Sciences of the United States of America*, 38(8):716.

Benda, J. and Herz, A. V. (2003). A universal model for spike-frequency adaptation. *Neural computation*, 15(11):2523–2564.

Bengio, Y. (2009). Learning deep architectures for A.I. *Foundations and trends in Machine Learning*, 2(1):1–127.

Benucci, A., Ringach, D. L., and Carandini, M. (2009). Coding of stimulus sequences by population responses in visual cortex. *Nature neuroscience*, 12(10):1317–24.

Berens, P., Ecker, A. S., Cotton, R. J., Ma, W. J., Bethge, M., and Tolias, A. S. (2012). A fast and simple population code for orientation in primate v1. *The Journal of Neuroscience*, 32(31):10618–10626.

Berens, P., Ecker, A. S., Gerwinn, S., Tolias, A. S., and Bethge, M. (2011). Reassessing optimal neural population codes with neurometric functions. *Proceedings of the National Academy of Sciences of the United States of America*, 108(11):4423–8.

Bertsekas, D. P. M. (2012). *Dynamic Programming and Optimal Control*. Athena Scientific, 4th edition.

Bethge, M., Rotermund, D., and Pawelzik, K. R. (2002). Optimal short-term population coding: When Fisher information fails. *Neural Computation*, 14(10):2317–2351.

Bobrowski, O., Meir, R., and Eldar, Y. C. (2009). Bayesian filtering in spiking neural networks: noise, adaptation, and multisensory integration. *Neural computation*, 21(5):1277–320.

Boerlin, M. and Denève, S. (2011). Spike-based population coding and working memory. *PLoS computational biology*, 7(2):e1001080.

Boyen, X. and Koller, D. (1998). Tractable inference for complex stochastic processes. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 33–42. Morgan Kaufmann Publishers Inc.

Brockwell, A., Rojas, A., and Kass, R. (2004). Recursive bayesian decoding of motor cortical signals by particle filtering. *Journal of Neurophysiology*, 91(4):1899–1907.

Brown, R. (1973). Integrated navigation systems and kalman filtering: a perspective. *Navigation*, 19(4):355–362.

Brunel, N. and Nadal, J. (1998). Mutual information, Fisher information, and population coding. *Neural Computation*, 10(7).

Brunel, N. and Nadal, J.-P. (1997). Optimal tuning curves for neurons spiking as a poisson process. In *ESANN*. Citeseer.

Bucy, R. S. (1965). Nonlinear filtering theory. *Automatic Control, IEEE Transactions*, 10(2):198.

Cadieu, C. F. and Olshausen, B. A. (2008). Learning Transformational Invariants from Natural Movies. In *Advances in Neural Information Processing Systems 21*, pages 1–8.

Conrad, W. and Corrado, C. (1979). Application of the kalman filter to revisions in monthly retail sales estimates. *Journal of Economic Dynamics and Control*, 1(2):177–198.

Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*, volume 6 of *Wiley Series in Telecommunications*. Wiley.

Crisan, D. and Doucet, A. (2002). A survey of convergence results on particle filtering methods for practitioners. *Signal Processing, IEEE Transactions on*, 50(3):736–746.

Curcio, C. A., Sloan, K. R., Packer, O., Hendrickson, A. E., and Kalina, R. E. (1987). Distribution of cones in human and monkey retina: individual variability and radial asymmetry. *Science*, 236(4801):579–582.

Dayan, P. and Abbott, L. F. (2001). *Theoretical neuroscience*. Cambridge, MA: MIT Press.

Deisenroth, M. P., Huber, M. F., and Hanebeck, U. D. (2009). Analytic moment-based gaussian process filtering. In *Proceedings of the 26th annual international conference on machine learning*, pages 225–232. ACM.

Dobzhansky, T. (1973). Nothing in Biology Makes Sense Except in the Light of Evolution. *The American Biology Teacher*, 35(3):125–129.

Doucet, A., De Freitas, N., and Gordon, N. (2001). An introduction to sequential monte carlo methods. In *Sequential Monte Carlo methods in practice*, pages 3–14. Springer.

Ergun, A., Barbieri, R., Eden, U. T., Wilson, M. A., and Brown, E. N. (2007). Construction of point process adaptive filter algorithms for neural systems using sequential Monte Carlo methods. *IEEE Transactions on Biomedical Engineering*, 54(3):419–428.

Faisal, A. A., Selen, L. P., and Wolpert, D. M. (2008). Noise in the nervous system. *Nature Reviews Neuroscience*, 9(4):292–303.

Ganguli, D. and Simoncelli, E. P. (2011). Implicit encoding of prior probabilities in optimal neural populations. (December 2010):6–9.

Gardiner, C. W. (2004). *Handbook of Stochastic Methods: for Physics, Chemistry and the Natural Sciences*, volume Vol. 13 of *Series in synergetics*. Springer.

Gerstner, W. and Naud, R. (2009). How good are neuron models? *Science*, 326(5951):379–380.

Gerwinn, S., Macke, J., and Bethge, M. (2009). Bayesian Population Decoding of Spiking Neurons. *Frontiers in computational neuroscience*, 3(October):14.

Gilbert, C. D. and Li, W. (2013). Top-down influences on visual processing. *Nature Reviews Neuroscience*, 14(5):350–363.

Guo, D., Shamai, S., and Verdú, S. (2005). Mutual information and minimum mean-square error in gaussian channels. *Information Theory, IEEE Transactions on*, 51(4):1261–1282.

Hahnloser, R. H. R. and Bla, F. (2011). An Efficient Coding Hypothesis Links Sparsity and Selectivity of Neural Responses. *October*, 6(10).

Häusler, C. and Susemihl, A. (2012). Temporal autoencoding restricted boltzmann machine. *arXiv preprint arXiv:1210.8353*.

Häusler, C., Susemihl, A., and Nawrot, M. P. (2013a). Natural image sequences constrain dynamic receptive fields and imply a sparse code. *Brain research*, 1536:53–67.

Häusler, C., Susemihl, A., Nawrot, M. P., and Opper, M. (2013b). Temporal autoencoding improves generative models of time series. *arXiv preprint arXiv:1309.3103*.

Hodgkin, A. L. and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500.

Hubel, D. H. and Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148(3):574.

Huys, Q. J. M., Zemel, R. S., Natarajan, R., and Dayan, P. (2007). Fast population coding. *Neural Computation*, 19(2):404–441.

Kandel, E. R., Schwartz, J. H., Jessell, T. M., et al. (2000). *Principles of neural science*, volume 4. McGraw-Hill New York.

Kappen, H. (2011). Optimal control theory and the linear Bellman equation. In Barber, D., Cemgil, A. T., and Chiappa, S., editors, *Bayesian Time Series Models*, chapter 1, pages 1–31. Cambridge university press, Cambridge, 1st edition.

Laughlin, S. (1981). A simple coding procedure enhances a neuron's information capacity. *Z. Naturforsch*, 36(c):910–912.

Lewicki, M. S. (2002). Efficient coding of natural sounds. *Nature neuroscience*, 5(4):356–63.

Ma, W. J., Beck, J. M., Latham, P. E., and Pouget, A. (2006). Bayesian inference with probabilistic population codes. *Nature neuroscience*, 9(11):1432–8.

MacKay, D. J. C. (2003). *Information theory, inference and learning algorithms*. Cambridge university press.

Malzahn, D. and Opper, M. (2005). A statistical physics approach for the analysis of machine learning algorithms on real data. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(11):P11001.

Merhav, N. (2011). Optimum estimation via gradients of partition functions and information measures: a statistical-mechanical perspective. *Information Theory, IEEE Transactions on*, 57(6):3887–3898.

Minka, T. P. (2001). Expectation propagation for approximate bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc.

Ø ksendal, B. (2003). *Stochastic Differential Equations: An Introduction with Applications*, volume 10 of *Universitext*. Springer.

Olshausen, B. A. and Field, D. J. (1996). Natural image statistics and efficient coding. *Network*, 7(2):333–339.

Opper, M. (1998). A bayesian approach to online learning. In *Online Learning in Neural Networks*, pages 363–378. Cambridge University Press.

Opper, M. and Winther, O. (2000). Gaussian processes for classification: Mean-field algorithms. *Neural Computation*, 12(11):2655–2684.

Park, M. and Pillow, J. W. (2011). Receptive Field Inference with Localized Priors. *PLoS Computational Biology*, 7(10):e1002219.

Pearson, J. B. and Stear, E. B. (1974). Kalman filter applications in airborne radar tracking. *Aerospace and Electronic Systems, IEEE Transactions on*, (3):319–329.

Pillow, J. W., Ahmadian, Y., and Paninski, L. (2011). Model-based decoding, information estimation, and change-point detection techniques for multineuron spike trains. *Neural Computation*, 23(1):1–45.

Pillow, J. W., Shlens, J., Paninski, L., Sher, A., Litke, A. M., Chichilnisky, E. J., and Simoncelli, E. P. (2008). Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, 454(7207):995–9.

Privault, N. (2014). *Stochastic Finance - An Introduction with Market Examples*. Chapman & Hall/CRC Financial Mathematical Series.

Ranganathan, A. (2004). Assumed density filtering, lecture notes. http://www.ananth.in/Notes_files/adf.pdf.

Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 1st edition.

Åström, K. J. (2006). *Introdution to Stochastic Control Theory*. Courier Dover Publications, Mineola, NY, 1st edition.

Schneidman, E., Still, S., Ii, M. J. B., and Bialek, W. (2003). Network Information and Connected Correlations. 1(December):3–6.

Sennewald, K. and Wälde, K. (2006). "Ito's Lemma" and the Bellman equation for Poisson processes: An applied view. *Journal of Economics*, 89(1):1–36.

Shannon, C. E. (1948). The mathematical theory of communication. 1963. *Bell System Technical Journal*, 27:379–423 and 623–656.

Snyder, D. L. (1972). Filtering and detection for doubly stochastic Poisson processes. *IEEE Transactions on Information Theory*, 18(1):91–102.

Susemihl, A., Meir, R., and Opper, M. (2011). Analytical Results for the Error in Filtering of Gaussian Processes. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P., Pereira, F., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, pages 2303–2311.

Susemihl, A., Meir, R., and Opper, M. (2013). Dynamic state estimation based on poisson spike trains: towards a theory of optimal encoding. *Journal of Statistical Mechanics: Theory and Experiment*, 2013(03):P03009.

Susemihl, A., Meir, R., and Opper, M. (2014). Optimal Population Codes for Control and Estimation. *ArXiv e-prints*.

Theodorou, E. and Todorov, E. (2012). Stochastic optimal control for nonlinear markov jump diffusion processes. *American Control Conference (ACC), 2012*.

Thorpe, S., Fize, D., and Marlot, C. (1996). Speed of processing in the human visual system. *Nature*, 381(6):520–522.

Tkacik, G., Prentice, J. S., Balasubramanian, V., and Schneidman, E. (2010). Optimal population coding by noisy spiking neurons. *Proceedings of the National Academy of Sciences of the United States of America*, 107(32):14419–24.

Urry, M. J. and Sollich, P. (2013). Random walk kernels and learning curves for gaussian process regression on random graphs. *Journal of Machine Learning Research*, 14:1801–1835.

Vicente, R., Susemihl, A., Jericó, J., and Caticha, N. (2014). Moral foundations in an interacting neural networks society: A statistical mechanics analysis. *Physica A: Statistical Mechanics and its Applications*.

Whittle, P. (1981). Risk-sensitive linear/quadratic/gaussian control. *Advances in Applied Probability*, pages 764–777.

Wu, W., Gao, Y., Bienenstock, E., Donoghue, J. P., and Black, M. J. (2006). Bayesian population decoding of motor cortical activity using a kalman filter. *Neural computation*, 18(1):80–118.

Wu, Y. and Verdú, S. (2010). Functional properties of mmse. In *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, pages 1453–1457. IEEE.

Yaeli, S. and Meir, R. (2010). Error-based analysis of optimal tuning functions explains phenomena observed in sensory neurons. *Frontiers in computational neuroscience*, 4(October):16.

Zakai, M. (1969). On the optimal filtering of diffusion processes. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 11(3):230–243.

Zhang, K. and Sejnowski, T. T. J. (1999). Neuronal tuning: To sharpen or broaden? *Neural Computation*, 11(1):75–84.

# A
# Appendix

## A.1 Maximum Entropy Distributions

Assume one has a random variable $X$ with a a finite set of outcomes $\mathscr{A}_x = \{x_i\}$, and one wishes to find the distribution over $X$ which maximizes the entropy

$$H[P_X] = \sum_{x_i} P_X(x_i) \log\left(\frac{1}{P_X(x_i)}\right).$$

One can use a Lagrange multiplier to enforce the normalisation of $P_X(x)$ and then take a derivative of the entropy with respect to $P_X$. This will lead to

$$\mathscr{L}[P_X, \beta] = H[P_X] - \beta\left(\sum_{x_i} P_X(x_i) - 1\right).$$

The derivative of $\mathscr{L}$ with respect to $P_X$ will then give

$$\frac{\delta \mathscr{L}}{\delta P_X(x_i)} = -\log(P_X(x_i)) + 1 - \beta.$$

Setting that to zero one obtains

$$P_X(x_i) = \exp(-1 + \beta).$$

This is a uniform distribution, as $\beta$ is a normalization constant and does not depend on $x$. Likewise, if one has any other information about the distribution, such as the expected value of some function of $X$, this can be included as a Lagrange multiplier as well. Generally, if one has a number of functions $f_j(x)$ whose expected value are known to be $e_j$, one can obtain a maximum entropy distribution similarly by writing

$$\mathscr{L}[P_X, \boldsymbol{\beta}] = H[P_X] - \beta_0\left(\sum_{x_i} P_X(x_i) - 1\right) + \sum_j \beta_j\left(\sum_{x_i} f_j(x_i) P_X(x_i) - e_j\right).$$

The derivative will then be given by

$$\frac{\delta \mathscr{L}}{\delta P_X(x_i)} = -\log(P_X(x_i)) + 1 - \beta_0 - \sum_j \beta_j f_j(x_i).$$

Which will lead to

$$P_X(x_i) = \exp\left(-1 + \beta_0 + \sum_j \beta_j f_j(x_i)\right).$$

Note that the values of every constant $\beta_j$ have to be determined so that the expected values of $f_j(x)$ match the known values. The Boltzmann distribution is given by this derivation if one requires the expected value of the energy of the system to be equal to some expected value, and its associated multiplier will be the inverse temperature $\beta = 1/k_B T$.

### *A.2 ADF and Moment Matching*

Say one wants to approximate some complex probability density $p(x)$ by a simpler parametric density $q(x)$. One alternative is to minimise the KL-divergence of the two densities

$$KL[p\|q] = \int dx p(x) \log \frac{p(x)}{q(x)}.$$

If one chooses $q(x)$ to be in the family of exponential distributions, $q(x)$ can be written as

$$q(x) = h(x) \exp\left[\phi^\top u(x) - g(\phi)\right],$$

where $\phi$ is called the vector of natural parameters and $u(x)$ is the vector of natural statistics. To minimise the KL-divergence, one needs to set its derivative with respect to the parameters of $q$ to zero. Therefore

$$\frac{dKL[p\|q]}{d\phi} = \int dx p(x) \left(u(x) - \frac{dg}{d\phi}\right). \qquad \text{(A.1)}$$

The derivative of $g$ can be seen to given the expected value of the natural statistics. Since $q$ is a density, one has

$$\int dx q(x) = 1, \text{ and therefore } \frac{d}{d\phi} \int dx q(x) = 0.$$

But

$$\frac{d}{d\phi} \int dx q(x) = \int dx \frac{dq}{d\phi} = \int dx q(x) \left(u(x) - \frac{dg}{d\phi}\right),$$

and therefore

$$\frac{dg}{d\phi} = \int dx u(x) q(x) = \boldsymbol{E}_q\left[u(x)\right].$$

Inserting this into equation (A.1), one gets

$$\frac{dKL[p\|q]}{d\phi} = \int dx p(x) u(x) - \boldsymbol{E}_q\left[u(x)\right] = 0,$$

and finally the minimum condition for $q$

$$\boldsymbol{E}_q[u(x)] = \int dx p(x) u(x).$$

This states that the distribution $q(x)$ that minimises the KL-divergence is the exponential distribution whose natural statistics match the ones of the distribution $p(x)$. This is often called a moment matching approach, as all the moments defined by the natural statistics of $q(x)$ will match the ones of $p(x)$.

This appendix follows lecture notes by Ananth Ranganathan.[1] If one took, for example, a Gaussian distribution, one would have $u(x) = (x, x^2)^\top$. The moment matching would lead one to match the mean and covariance of $p(x)$ to the mean and covariance of our approximating distribution $q(x)$. This is very practical as it allows one to bypass any optimisation procedure to determine the parameters of $q(x)$. However, one still needs to evaluate the moments of the possibly intractable density $p(x)$, which might not be feasible. One alternative to this is to estimate the moments by a sampling procedure.

[1] Ranganathan, A. (2004). Assumed density filtering, lecture notes. http://www.ananth.in/Notes_files/adf.pdf

### *A.3 Solving the Mean-field Kernel Integral Equation*

In chapter 3 I have derived an integral equation for the mean-field approximation of the posterior kernel. The mean-field posterior kernel $g(u) = \boldsymbol{E}_{mf}[X(t+u)X(t)]$ of a process $X(t)$ with prior distribution given by a GP with kernel $k$ observed by Poisson spikes with frequency $\hat{\lambda}$ and tuning width $\alpha$, obeys the integral equation

$$g(u) = k(u,0) - \frac{\hat{\lambda}}{\alpha^2 + g(0)} \int_0^\infty g(s+u)g(s)ds.$$

To obtain a numerical approximation for $g$ I can simply guess an initial value of $g$ and insert that in the right-hand side to obtain an improved guess. This can then be repeated until it converges, for example by a squared-distance criterion. This is the fix-point method, though one needs to be careful to choose the starting condition and iteration rules to be sure to converge. The simplest approach to iterating equation (3.19) is to choose a cutoff $D$, after which the value of the integrand can be ignored. After that, one needs to choose a numerical integration method to evaluate the remaining integral. This leads to the iteration

$$g^{i+1}(u) = k(u,0) - \frac{\hat{\lambda}}{\alpha^2 + g^i(0)} \int_0^D g^i(s+u)g^i(s)ds.$$

Taking $g^0(u) = k(u,0)$, the prior kernel, and using the parallelogram integration method, will lead to the results shown in

figure 3.5. One can then establish a stopping time by a tolerance in the mean-squared distance between two consecutive iterations

$$d(g^{i+1}, g^i) = \int_0^D (g^{i+1}(u) - g^i(u))^2 \, du.$$

For figure 3.5 I have used a tolerance of $10^{-10}$ and have taken a cutoff $D = 12$.

### *A.4 Deriving the PDE for $f(\Sigma, t)$*

To obtain an equation for $f(\Sigma, t)$ I need to evaluate the averages $\boldsymbol{E}_{X(t)}[\lambda^m(X(t))]$. This is simply the average of a squared exponential under a Gaussian measure. Here I will assume $E$ is invertible, leading to

$$\boldsymbol{E}_{X(t)}[\lambda^m(X(t))] = \frac{\phi}{\sqrt{|I + \Sigma E^{-1}}} \exp\left[-\frac{1}{2}(\mu - \theta_m)^\top (\Sigma + E)^{-1}(\mu - \theta_m)\right].$$

It is now straightforward to evaluate the sum over all neurons in the HJB equation. This yields

$$\sum_m \boldsymbol{E}_{X(t)}[\lambda^m(X(t))]\left[(\mu + \Delta^m\mu)^\top S(t)(\mu + \Delta^m\mu) + f(\Sigma + \Delta\Sigma, t) - \mu^\top S(t)\mu - f(\Sigma(t), t)\right].$$

By considering the sum over neurons as a Gaussian integral over $\theta$, I can rewrite the $\mu$-dependent terms approximately as

$$\frac{1}{|\Delta\theta|^N \sqrt{|I + \Sigma E^{-1}|}} \int d\theta e^{-(\mu-\theta)^\top (\Sigma+E)^{-1}(\mu-\theta)/2}\left(\Delta\mu^\top S(t)\mu + \Delta\mu^\top S(t)\Delta\mu\right),$$

where $\Delta\theta$ is the distance between neighbouring neurons, and $N$ is the dimension of the stimulus space were $\mu$ resides in. Since $\Delta\mu$ is a linear function of $\mu - \theta$, the first term will be zero. The second term will give

$$\int d\theta e^{-(\mu-\theta)^\top (\Sigma+E)^{-1}(\mu-\theta)/2}(\mu-\theta)^\top(\Sigma+E)^{-1}\Sigma S(t)\Sigma(\Sigma+E)^{-1}(\mu-\theta) = (2\pi)^{N/2}\sqrt{|\Sigma+E|}\,\mathrm{Tr}[\Sigma S(t)\Sigma(\Sigma+E)^{-1}] \quad \text{(A.2)}$$

The full expectation of the jump term will therefore give

$$\sum_m \boldsymbol{E}_{X(t)}[\lambda^m(X(t))]\left[(\mu + \Delta^m\mu)^\top S(t)(\mu + \Delta^m\mu) + f(\Sigma + \Delta\Sigma, t) - \mu^\top S(t)\mu - f(\Sigma(t), t)\right] = \frac{(2\pi)^{N/2}\phi\sqrt{|E|}}{|\Delta\theta|^N}\left[f(\Sigma + \Delta\Sigma, t) - f(\Sigma, t) + \mathrm{Tr}[\Sigma S(t)\Sigma(\Sigma + E)^{-1}]\right]. \quad \text{(A.3)}$$

Note that the pre factor of the final term is exactly the population firing rate $\hat{\lambda} = \sum_m \lambda^m(x)$. Inserting this back into the HJB equation and collecting the terms for $f$ one obtains

$$-\frac{\partial f}{\partial t} = \mathrm{Tr}\left(Q(t)\Sigma\right) + \mathrm{Tr}\Big[\tfrac{\partial f}{\partial \Sigma}\left(A\Sigma + \Sigma A^\top + H\right)\Big] \qquad \text{(A.4)}$$
$$+ \quad \hat{\lambda}\Big[f(\Sigma+\Delta\Sigma,t) - f(\Sigma,t) + \mathrm{Tr}\Big(\Sigma S(t)\Sigma\left(\Sigma+E\right)^{-1}\Big)\Big].$$

### *A.5 Solving for the uncertainty cost $f(\Sigma,t)$*

The uncertainty costs of a LQG control system observed through spike trains from a dense Gauss-Poisson neural population obeys the PDE

$$-\frac{\partial f}{\partial t} = \mathrm{Tr}\left(Q(t)\Sigma\right) + \mathrm{Tr}\Big[\tfrac{\partial f}{\partial \Sigma}\left(A\Sigma + \Sigma A^\top + H\right)\Big]$$
$$+ \quad \hat{\lambda}\Big[f(\Sigma+\Delta\Sigma,t) - f(\Sigma,t) + \mathrm{Tr}\Big(\Sigma S(t)\Sigma\left(\Sigma+E\right)^{-1}\Big)\Big],$$

with boundary condition $f(\Sigma,T) = \mathrm{Tr}\left(\Sigma Q_T\right)$. This is a very cumbersome equation, and most approaches would be inapplicable due to the jump terms present. The application of numerical solutions through a discretisation of time and $\Sigma$-space is also not straightforward, as the nonlinear nature of the jumps, would force one to estimate the value of the function $f$ at a large number of points outside of the discretisation grid. A simple solution can, however, be derived via the Feynman-Kac formula. The Feynman-Kac formula allows one to write the solution of a PDE as an expectation over paths of a stochastic process. Given a parabolic PDE

$$\frac{\partial u}{\partial t} + \mu(x,t)\frac{\partial u}{\partial x} + \frac{1}{2}\sigma(x,t)^2\frac{\partial^2 u}{\partial t^2} - V(x,t)u(x,t) + f(x,t) = 0,$$

the Feynman-Kac formula says that the solution $u(x,t)$ with boundary condition $u(x,T) = \phi(x)$ can be written as a conditional expectation over paths of a stochastic process, given by

$$u(x,t) = \boldsymbol{E}_X\left[\int_t^T e^{-\int_t^r V(X(s),s)ds} f(X(r),r)dr + e^{-\int_t^T V(X(s),s)ds}\phi(X(T)) \,\middle|\, X(t) = x\right],$$

where the expectation is over paths of the process given by

$$dX(t) = \mu(X(t),t)dt + \sigma(x,t)dW(t).$$

This can be extended to general processes with jumps as well, and I will give a short derivation of this result below.

In the present case, I will take the process

$$d\Sigma(t) = (A\Sigma(t) + \Sigma(t)A^\top + H)dt + \Delta\Sigma(t)dN(t). \quad \text{(A.5)}$$

This is exactly the dynamics of the covariance from the Point process filter used to estimate the system's state. Define, then

$$Y(t) = f(\Sigma(t), t) - \int_t^T \left[\mathrm{Tr}\left(Q(t)\Sigma(u)\right) + \bar{\lambda}\,\mathrm{Tr}\left(\Sigma(u)S(u)\Sigma(u)(\Sigma(u)+E)^{-1}\right)\right] du,$$

where $f(\Sigma, t)$ is a solution to equation (4.15). I will below show that $Y(t)$ is a martingale, allowing me to write the value of $f(\Sigma, t)$ as an average over paths of the stochastic process equation (A.5). The variation of the process $Y$ will be given by

$$dY(t) = df + \left[\mathrm{Tr}\left(Q(t)\rho(t)\right) + \hat{\lambda}\,\mathrm{Tr}\left(\Sigma(t)S(t)\Sigma(t)(\Sigma(t)+E)^{-1}\right)\right] dt.$$

Via the Itō Lemma, we have

$$df = \left(\frac{\partial f}{\partial t} + \mathrm{Tr}\left[\frac{\partial f}{\partial \Sigma}\left(A\Sigma + \Sigma A^{\top} + H\right)\right] + \hat{\lambda}\Delta f(t)\right) dt + dJ_f(s),$$

where

$$\Delta f(t) = f(\Sigma(t) + \Delta\Sigma(t), t) - f(\Sigma(t), t),$$

is the jump incurred in $f$ when there is a jump in $\Sigma(t)$ and $dJ_f(s)$ is the process

$$dJ_f(s) = dN(t)\Delta f_s - \hat{\lambda}\Delta f_s dt,$$

where $\boldsymbol{E}[dJ_f(s)] = 0$. This leads to

$$dY(t) = \left(\frac{\partial f}{\partial t} + \mathrm{Tr}\left[\frac{\partial f}{\partial \Sigma}\left(A\Sigma + \Sigma A^{\top} + H\right)\right] + \hat{\lambda}\Delta f + \mathrm{Tr}\left(Q(t)\rho(t)\right) + \bar{\lambda}\,\mathrm{Tr}\left(\Sigma(t)S(t)\Sigma(t)(\Sigma(t)+E)^{-1}\right)\right)$$

The term in parentheses is zero, as $f$ is a solution to equation (4.15). Therefore, integrating $dY(t)$ from $t$ to $T$, I obtain

$$Y(T) = Y(t) + \int_t^T dJ_f(u).$$

Taking the average with respect to the paths of process $\Sigma(t)$ then leads to

$$\boldsymbol{E}\left[Y_T | \Sigma(t) = \Sigma\right] = \boldsymbol{E}\left[Y(t) | \Sigma(t) = \Sigma\right] + \boldsymbol{E}\left[\int_t^T dJ_f(u) \,\middle|\, \rho(t) = \Sigma\right] = \boldsymbol{E}\left[Y(t) | \Sigma(t) = \Sigma\right],$$

where in the last step I have used that $\boldsymbol{E}[dJ_f(s)] = 0$. This shows that $Y(t)$ is a Martingale. This leads to the Feynman-Kac formula for $f$

$$\begin{aligned} f(\Sigma, t) = {} & \boldsymbol{E}[f(\Sigma(T), T) | \Sigma(t) = \Sigma] \\ & + \boldsymbol{E}\left[\int_t^T \left[\mathrm{Tr}\left(Q(t)\Sigma(u)\right) + \bar{\lambda}\,\mathrm{Tr}\left(\Sigma(u)(\Sigma(u)+E)^{-1}\Sigma(u)S(u)\right)\right] du \,\middle|\, \rho(t) = \Sigma\right]. \end{aligned}$$

The evolution of $\boldsymbol{E}\left[\Sigma(t)\right]$ is given by

$$\frac{\partial \boldsymbol{E}\left[\Sigma(t)\right]}{\partial t} = A\boldsymbol{E}\left[\Sigma(t)\right] + \boldsymbol{E}\left[\Sigma(t)\right]A^{\top} + H - \bar{\lambda}\boldsymbol{E}\left[\Sigma(t)(\Sigma(t)+A)^{-1}\Sigma(t)\right],$$

therefore, one can use this expression to directly estimate the trace average in the equation for $f$. This yields

$$\hat{\lambda}\,\mathrm{Tr}\Big(\boldsymbol{E}\left[\Sigma(t)(\Sigma(t)+A)^{-1}\Sigma(t)\right]S(t)\Big) = \mathrm{Tr}\left[\left(\frac{\partial \boldsymbol{E}\left[\Sigma(t)\right]}{\partial t} + A\boldsymbol{E}\left[\Sigma(t)\right] + \boldsymbol{E}\left[\Sigma(t)\right]A^{\top} + H\right)S(t)\right].$$

This prevents one from having to calculate expensive matrix inversions and allows one to write

$$\begin{aligned} f(\Sigma,t) = {} & \mathrm{Tr}\Big(\boldsymbol{E}^t_{\Sigma}[\Sigma(T)]Q_T\Big) \\ + {} & \int_t^T \left[\mathrm{Tr}\Big((Q(u)+S(u)A+A^{\top}S(u))\boldsymbol{E}^t_{\Sigma}(\Sigma(u))\Big) + \mathrm{Tr}\left(HS(u)\right) - \mathrm{Tr}\left(\tfrac{\partial \boldsymbol{E}^t_{\Sigma}(\Sigma(u))}{\partial u}S(u)\right)\right]du, \end{aligned}$$

where I have written

$$\boldsymbol{E}^t_{\Sigma}(X) = \boldsymbol{E}[X|\Sigma(t)=\Sigma],$$

and used the boundary condition for $f$. By linearity of the trace operator and integration by parts one has

$$\mathrm{Tr}\left(\int_t^T \frac{\partial \boldsymbol{E}^t_{\Sigma}(\Sigma(u))}{\partial u}S(u)du\right) = \mathrm{Tr}(\boldsymbol{E}^t_{\Sigma}(\Sigma(u))S(u)\big|_t^T) - \mathrm{Tr}\left(\int_t^T \boldsymbol{E}^t_{\Sigma}(\Sigma(u))\dot{S}(u)du\right).$$

$\dot{S}$ in turn is given by the Riccatti equation, leading to

$$\begin{aligned} \mathrm{Tr}\left(\int_t^T \frac{\partial \boldsymbol{E}^t_{\Sigma}(\Sigma(u))}{\partial u}S(u)du\right) = {} & \mathrm{Tr}(\boldsymbol{E}^t_{\Sigma}(\Sigma(u))S(u)\big|_t^T) \\ + {} & \mathrm{Tr}\Big(\textstyle\int_t^T \boldsymbol{E}^t_{\Sigma}(\Sigma(u))(Q(u)+S(u)A+A^{\top}S(u)-S(u)B^{\top}R(u)^{-1}BS(u)du\Big). \end{aligned}$$

This leads finally to the expression of the uncertainty related costs for the control problem at hand:

$$f(\Sigma,t) = \mathrm{Tr}\left(\Sigma(t)S(t)\right) + \int_t^T \mathrm{Tr}\left(HS(u)\right)du + \int_t^T \mathrm{Tr}\Big(S(u)B^{\top}R(u)^{-1}BS(u)\boldsymbol{E}^t_{\Sigma}(\Sigma(u))\Big)du. \tag{A.6}$$

To solve numerically for $f(\Sigma,t)$ one can now simply take a large number of paths from the $\Sigma(t)$ process and average the integral over many realisations. Alternatively, one could approximate the dynamics of $\boldsymbol{E}^t_{\Sigma}(\Sigma(u))$, for example with a mean-field approximation, and use that approximate dynamics to evaluate $f$.