Chapter 5

Non-parametric Bayesian State Space Estimator

In both Chapters 3 and 4, we demonstrated that it is feasible to learn a POMDP policy from human teachers. Further, by adding a simple binary reward function we were able to take into consideration the quality of these demonstrations. provided by the teachers. With this, we showed that our Reinforcement Learning extension, RL-PbD-POMDP was able to yield improved policies even when provided with a limited number of demonstrations taken from the worst teachers.

Both tasks from the previous two chapters (search for wooden block on a table and peg-in-hole) fall into the category of goal oriented **active-localisation**. In general, the localisation problem consists of estimating position parameters given noisy observations whereas active-localisation refers to a policy which actively takes actions to acquire information to decrease the uncertainty of the position estimate. In localisation, the model of the world also known as the **map** is considered **prior knowledge**. This assumption constrains localisation to an environment in which schematics exist and can be used as the world model such as in the case of offices and buildings. If the map is not known a priori, then Simultaneous Localisation And Mapping (**SLAM**) algorithms have to be used instead of localisation. Typically, the map consists of a set of features also known as landmarks which can be identified by sensors, and SLAM algorithms maintain a filtered joint probability distribution over both the agent's and features' position which is updated in accordance to a generic Bayesian State Space Filter (BSSF) (see Figure 2.5 on page 20).

In this Chapter, we consider an agent tasked with searching for a set of objects on a *Table* world (see Figure 5.1), in which exteroceptive feedback is extremely limited. The agent can only sense an object after making physical contact with it (bumping into it). The agent's uncertainty of its location and that of the object is encoded by probability distributions $P(\cdot)$, which at initialisation are known as the agent's prior beliefs.

Figure 5.1 illustrates a particular instance of the agent's beliefs. The agent is currently located in the bottom table and has only a limited knowledge of its location, somewhere near the right edge of the table.

As the agent explores the world, it integrates all sensing information at each time step and updates its prior beliefs to posteriors (the result of the prior belief

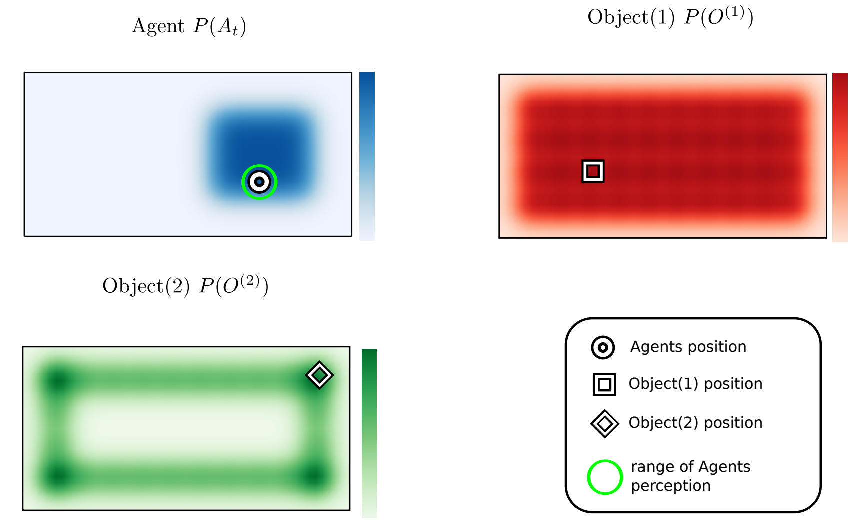

Figure 5.1: *Table Environment* Table World (delimited by the black rectangle), viewed from above, and the agent's beliefs. There are three different probability density functions present on the table. The blue represents the believed location of the agent, the red and green probability distributions are associated with object 1 and 2. The white shapes in each figure represent the true location of each associated object or agent.

after integrating motion and sensory information). All current SLAM methods are limited in that they only consider uncertainty induced by sensing inaccuracy inherent in the sensor and motion models. In our setting as the sensory information is strictly haptic, we can confidently assume no measurement noise. In the search task illustrated in Figure 5.1, the beliefs and sparse measurement information available to the agent are the source of the uncertainty which is, the absence of positive object measurements. This is known as **negative information** (Thrun et al., 2005b, p.313) Thrun (2002); Hoffman et al. (2005). Thus SLAM methodologies which use the **Gaussian error** between the predicted and estimated position of features, such as in the case of EKF-SLAM and Graph-SLAM, will not work well in this setting.

The EKF SLAM algorithm, [...], can only process positive sightings of landmarks. It cannot process negative information that arises from the absence of landmarks. -Probabilistic Robotics (Thrun et al., 2005b, p.313)-

In addition to the negative sensing information, the original beliefs depicted in Figure 5.1 are **non-Gaussian** and **multi-modal**. We make **no assumption** regarding the form the beliefs can take. This implies that the joint distribution can no longer be parameterised by a Multivariate Gaussian. This is an assumption made in many SLAM algorithms, notably EKF-SLAM, and allows for a closed form solution to the state estimation problem. Without the Gaussian assumption no closed form solution to the filtering problem is feasible. Using standard non-parametric methods (Kernel Density, Gaussian Process, Histogram,...)

Attributes & Assumptions

- Non-Gaussian joint distribution, no assumptions are made with respect to its form.
- Mostly negative information available (absence of positive sightings of the landmarks).
- Joint distribution volume grows exponentially with respect to the number of objects and states.
- Joint distribution volume is dense, there is a lot of uncertainty.

Figure 5.2: Assumptions and attributes which have to be fulfilled by our Bayesian State Space Filter.

to represent or estimate the joint distribution becomes unrealistic after a few dimensions or additional map features. FastSLAM could be a potential candidate, however as it parameterises the position uncertainty of the agent by a particle filter and each particle has its own copy of the map, the memory demand will become significant. For planning purposes we would also want to have a single representation of the marginals. Figure 5.2 summarises the desirable attributes and assumptions for our filter.

The main contribution of our work and the importance to the field of Artificial Intelligence

An accurate estimate of the agent's belief space is a necessary precondition before planning or reasoning can be carried out. In a wide range of Artificial Intelligence (AI) applications the agent's beliefs are discrete. This non-parametric representation is the most unconstraining but comes at a cost. The parameterisation of the belief's joint distribution grows at the rate of a double exponential. We propose a Bayesian state estimator which delivers the same filtered beliefs as a traditional filter but without explicitly parametrising the joint distribution. Through the memorising of the measurement likelihood functions having been applied on the joint distribution and by taking advantage of their structure, we achieve a filter which grows linearly as opposed to exponentially in both time and space complexity. We refer to our novel filter as the Measurement Likelihood Memory Filter (MLMF) it keeps track of the history of measurement likelihood functions, referred to as the memory, which have been applied on the joint distribution. The MLMF filter allows to efficiently process negative information. To the authors knowledge there has been little research on the integration of negative information in a SLAM setting. Previous work considered the case of active localisation Hoffmann et al. (2006). The incorporation of negative information is useful in many contexts and in particular in Bayesian Theory of Mind Bake et al. (2011), where the reasoning process of a human is inferred from a Bayesian Network and in our own work de Chambrier and

Billard (2013) where we model the search behaviour of a intentionally blinded human. In such a setting much negative information is present and an efficient belief filter is required. Our MLMF is thus applicable to the SLAM & AI community in general and to the cognitive community which models human or agent behaviours through the usage of Bayesian state estimators.

By using this new representation we implement a set of passive search trajectories through the state space and demonstrate, for a discretised state space, that our novel filter is optimal with respect to the Bayesian criteria (the successive filtered posteriors are exact and not an approximation with respect to Bayes rule). We provide an analysis of the space and time complexity of our algorithm and prove that it is always more efficient even when considering worst case scenarios. Lastly we consider an Active-SLAM setting and evaluate the effect of how constraining the size of the number of memorised likelihood functions impacts the decision making process of a greedy one-step look-ahead planner.

5.1 Outline

The remaining part of this Chapter is structured as follows:

- 5.2 Background: Review of three prominent SLAM algorithms and their assumptions and an overview of active-localisation and exploration methods used with SLAM.
- 5.3 Bayesian State Space Estimation: Introduction to EKF-SLAM and why it is not suitable when mostly negative information is available. Description of the Histogram-SLAM algorithm and the assumptions which can be exploited.
- 5.4 Measurement Likelihood Memory Filter: Mathematical derivation of the MLMF, time and space complexity evaluation and extension to the scalable-MLMF.
- 5.5 Evaluation We numerically evaluate the time complexity of the scalable-MLMF and check its assumptions. We investigate the filter's sensitivity with respect to its parameters in an Active-SLAM setting.
- 5.6 Conclusion
- 5.7 Appendix

5.2 Background

5.2.1 SLAM

Estimating the location or state parameters of a mobile agent whilst simultaneously building a map of the environment has been regarded as one of the most important problems to be solved for agents to achieve true autonomy. It is a necessary precondition for any agent to have an estimation of the world at its disposal which accurately encompasses all knowledge and related uncertainties. There has been much research surrounding the field of Simultaneous Localisation And Mapping (SLAM) which branches out into a wide variety of sub-fields dealing with problems from building accurate noise models of the agent sensors Plagemann et al. (2007), to determining which environmental feature caused a particular measurement, also known as the data association problem Montemerlo and Thrun (2003) and many more.

Although the amount of research might seem overwhelming at first view, all current SLAM methodologies are founded on a single principle; the uncertainty of the map is correlated through the agent's measurements. When an agent localises itself (by reducing position uncertainty) all previously landmarks have their uncertainty reduced since the uncertainty is correlated with that of the agent's uncertainty.

There are three main paradigms to solving the SLAM problem. The first is EKF-SLAM (Extendend-Kalman Filter) Durrant-Whyte and Bailey (2006). EKF-SLAM models the full state, being the agent's parameters and environmental features, by a Multivariate Gaussian distribution. The uncertainty of each individual feature is parametrised by a mean (expected position of the feature) and covariance (how much uncertainty there is about the position of the feature).

The second approach is Graph-SLAM Grisetti et al. (2010). Graph-SLAM estimates the full path of the agent and considers every measurement to be a constraint on the agent's path. It is parameterised by the canonical Multivariate Gaussian. At each time step a new row and column is added to the precision matrix which encodes landmarks which have been observed as constraints on the robot's position. At predetermined times, a nonlinear sparse optimisation is solved to minimise all the accumulated constraints on the robot's path.

The third method is FastSLAM Montemerlo et al. (2003). FastSLAM exploits the fact that if we know the position of the agent with certainty all landmarks become independent. It models the distribution of the agent's position by a particle filter. Each particle has its own copy of the map and updates all landmarks independently which is the strength of this method. However, if many particles are required each must have its own copy of the map. It is beyond the scope of this chapter to provide a detailed review of these three paradigms and the reader is referred to Thrun et al. (2005b), Thrun and Leonard (2008).

5.2.2 Active-SLAM & Exploration

Active-SLAM refers to a decision theoretic process of choosing control actions so as to actively increase the convergence of the map. It is used in conjunction with exploration of an unknown environment in a SLAM setting. The two steps of this process are: (i) generate a set of candidate destination positions, (ii) evaluate these positions based on a utility function. The utility is a trade off between reducing the uncertainty of the map or reducing the uncertainty of the agent's position.

Most approaches use a two-level representation of the map in an exploration setting. At the lower level there is the chosen (landmark-based) SLAM filter and at the higher level a coarser representation of the world. Such representations can be occupancy grids Thrun and Bü (1996) which encode either occupied and free space or a topological representation Kollar and Roy (2008).

Early and current approaches to selecting candidate exploratory locations are based on evaluating Next-best-view González-Baños and Latombe (2002) locations. Next-best-view points are sampled around *free edges* which are at the horizon of the known map (*frontier* regions). In such a setting only target points are generated, not the full trajectory. Probabilistic Road Map (PRM) Kavraki et al. (1996) based methods have been used as planners to reach desired target locations, such as in Huang and Gupta (2008), where a Rapidly Exploring Random Trees (RRT) is combined with FastSLAM. In Valencia et al. (2012), paths to *frontier* regions are computed via PRM on a occupancy grid map and at the lower level they use Pose-SLAM (synonym for Graph-SLAM).

An alternative approach taken to generating candidate locations is the specification of high level macro actions, they being either *exploratory* or *revisiting* actions as is the case in Stachniss et al. (2005). Macro actions reduce the costly evaluation of actions, especially in the case of FastSLAM, which requires propagating the filter forward in time so as to infer the information gain of each action.

The last approach is to solve the planning problem through formulating it as Partially Observable Markov Decision Process (POMDP) Ross et al. (2008). However all methods take an approximation of the POMDP and consider a one time step planning horizon (Lidoris, 2011, p.37).

There are many ways of generating actions or paths, however their utility is nearly all exclusively based on the *information gain*, which is the estimated reduction of entropy a particular action or path would achieve. A few utilities use f-measures such as the Kullback-Leibler divergence. Evaluation of different utility metrics are presented in Carrillo et al. (2012); Tovar et al. (2006); Carlone et al. (2010).

5.3 Bayesian State Space Estimation

Bayesian State Space Estimation (BSSE) focuses on incorporating observa-

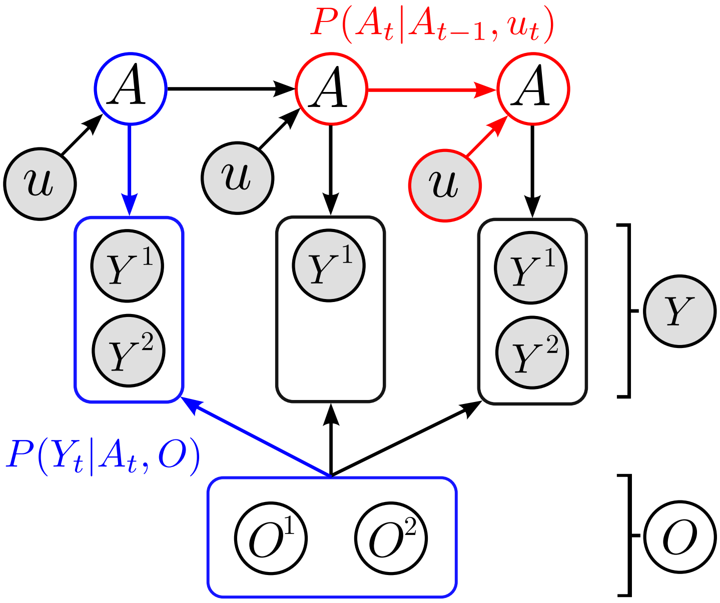

Figure 5.3: Directed graphical model of dependencies between the agent(A) and object(O)'s estimated location. Each object, $O^{(i)}$ is associated with one sensing random variable $Y^{(i)}$. The overall sensing random variable is $Y = \left[Y^{(1)},\ldots,Y^{(M-1)}\right]^{\mathrm{T}}$, where M is the total number of agent and object random variables in the filter. For readability we have left out the time index t from A and Y. Since the objects are static, they have no temporal process associated with them thus they will never have a time subscript. The two models necessary for filtering are the motion model $P(A_t|A_{t-1},u_t)$ (red) and measurement model $P(Y_t|A_t,O)$ (blue).

tions to update a prior distribution over the state space to a posterior distribution through the application of Bayes probability rules. The agent's random variable, A, is associated with the uncertainty of its location in the world, the same holds for the object(s') random variable(s), O. Given a sequence of actions and observations, $\{u_{1:t}, y_{0:t}\}$ (subscript $0:t$ is the set from the time $t=0$ to the current time, $t=t$), algorithms of the BSSE family incorporate this information to provide an estimate:

$$P(A_t, O|Y_{0:t}, u_{1:t}) \tag{5.3.1}$$

This is known as the filtering problem where all past information is incorporated to estimate the current state.

In Figure 5.3 we depict the general Bayesian Network (BN) of a BSSE. The BN conveys two types of information, the dependence and independence relations between the random variables in the graph which can be established through *d-separation* Shachter (1998), see Figure 5.4. Any joint probability distribution whose factorisation respects the structure of a BN is guaranteed to satisfy all the conditional independence statements which can be read from the graph, but the converse with respect to the dependence statements is not guaranteed (Barber, 2012, p.43).

Dependence & Independence

Conditional independence	
1) $A_{t+1} \perp\!\!\!\perp A_{t-1} \vert A_t$	First order Markov property.
2) $A_t \perp\!\!\!\perp Y_{t+1} \vert A_{t+1}$	Past states do not depend on future observations.
3) $A \perp\!\!\!\perp O \vert \emptyset$	Agent and object random variables are independent given no observation.
Conditional dependence	
1) $A \top\!\!\!\top O \vert Y$	Agent and object random variables will interact with each other given an observation

Figure 5.4: Dependence and independence relation between the random variables of the BN Figure 5.3

The **conditional dependence** $A \top\!\!\!\top O|Y$ is key to all BSSE and SLAM algorithms. The strength of the dependence between the agent and object random variable is governed by the measurement likelihood $P(Y_t|A_t, O)$. If it does not change the joint distribution then the agent and object random variables will be independent, $A \perp\!\!\!\perp O$. If they are independent, then no information acquired by the agent can be used to infer changes in the object estimates.

We next demonstrate the behaviour of the two different parameterisations of the joint distribution of the BN, Figure 5.3, in the case of the absence of direct sighting of the object by the agent. We first consider a Multivariate Gaussian parameterisation of the joint distribution, which is known as EKF-SLAM, and a different approach which discretises the joint distribution, called Histogram-SLAM.

EKF-SLAM

In EKF-SLAM the joint density $p(A_t, O|Y_{0:t}, u_{1:t}) = g(x; \mu_t, \Sigma_t)$ is parametrised by a single Gaussian function g with mean, $\mu_t = [\mu_{A_t}, \mu_{O^{(1)}}, \ldots, \mu_{O^{(M-1)}}]^{\mathrm{T}} \in \mathbb{R}^{3+2\cdot(M-1)}$ where the random variables are in $\mathbb{R}^2$, and covariance, Σ_t. The mean value of the agent $\mu_a = [x, y, \phi]^{\mathrm{T}} \in \mathbb{R}^3$ and those of the objects are $\mu_{O^{(i)}} = [x, y]^{\mathrm{T}} \in \mathbb{R}^2$.

$$\Sigma_t = \begin{bmatrix} \Sigma_a & \Sigma_{ao} \\ \Sigma_{oa} & \Sigma_o \end{bmatrix} \in \mathbb{R}^{(3+2\cdot(M-1))\times(3+2\cdot(M-1))} \tag{5.3.2}$$

The j'th object measurement is described by range and bearing $Y_t^{(j)} = [r, \phi]$ in the frame of reference of the agent, see Figure 2.4 page 19 for an illustration of a measurement update process. EKF-SLAM assumes that the measurement is corrupted by Gaussian noise, $\epsilon \sim \mathcal{N}(0, R)$, and this results in a measured

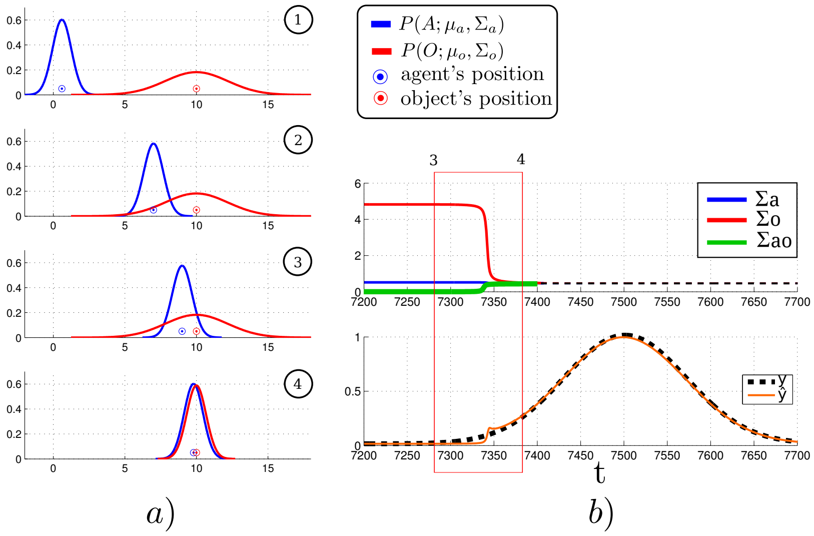

Figure 5.5: **a)** EKF-SLAM time slice evolutions of the pdfs. The temporal ordering is given by the numbers in the top right corner of each plot. The blue pdf represents the agent's believed location and the circle is the agent's true location. The same holds for the red distribution which represents the agent's belief of the location of an object. **b)** Evolution of the covariance components of Σ over time and true Y_t and expected measurements, $\hat{Y}_t$. Σ_a and Σ_o are the variances of the agent and object positions and Σ_{ao} is the cross-covariance term.

likelihood function of the following form:

$$p(Y_t|A_t, O_t) = \frac{1}{|2\pi R|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(Y_t - \hat{Y}_t)^{\mathrm{T}} R^{-1} (Y_t - \hat{Y}_t)\right) \tag{5.3.3}$$

$$\hat{Y}_t = \exp\left(-\frac{1}{2\sigma^2}||A_t - O||^2\right) \tag{5.3.4}$$

where the covariance, R, encompasses the uncertainty in the measurement and Equation 5.3.4 is the measurement function. The elements of the covariance matrix capture the measurement error between the true Y and expected $\hat{Y}$ range and bearing of the object. As the joint distribution is parametrised by a single Multivariate Gaussian, a closed form solution to the filtering Equations exists, called the Kalman Filter Durrant-Whyte and Bailey (2006).

The error between the true and expected measurement $e = (Y_t - \hat{Y}_t)$ is an important part of the application of EKF-SLAM. In our scenario the agent can only perceive the objects once he enters in direct contact with them. This means that the variance of the observation Y_t will be very low and will always be equal to $\hat{Y}$ until a contact occurs. To illustrate the problems which this gives rise to, we give an illustration of a 1D search. Figure 5.5 shows the resulting updates of the beliefs for 4 chosen time segments.

As expected we do not get the desired behaviour, which is that the beliefs start updating as soon as they are overlapping, see 2nd-3rd temporal snapshot in the Figure. Even when most of the belief mass of the agent's location pdf overlaps that of the object pdf, no belief update occurs. The multivariate Gaussian parameterisation only guarantees a dependency between the agent and object random variables when there is a positive sighting of the landmarks. This can been seen in Figure 5.5 (b), where the component Σ_{ao} is 0 most of the time which implies that $A \perp\!\!\!\perp O|Y$ which is undesirable. This confirms that the dependencies present in the structure given by the BN are dependent on the chosen parametrisation.

Histogram-SLAM

In Histogram-SLAM, the joint distribution is discretized and each bin has a parameter, $P(A_t = i, O = j|Y_{0:t}, u_{1:t}) = \boldsymbol{\theta}^{(ij)}$, which sums to one, $\sum_{ij} \boldsymbol{\theta}^{(ij)} = 1$. For shorthand notation we will write $P(A_t, O|Y_{0:t}, u_{1:t})$ instead of $P(A_t = i, O = j|Y_{0:t}, u_{1:t})$. The probability distribution of the agent's position is given by marginalising the object random variable:

$$P(A_t|Y_{0:t}, u_{1:t}) = \sum_{j=1}^{|O|} P(A_t, O|Y_{0:t}, u_{1:t}) \tag{5.3.5}$$

$$P(A_t = 1|Y_{0:t}, u_{1:t}) = \sum_{j=1}^{|O|} \boldsymbol{\theta}^{(1j)} \tag{5.3.6}$$

The converse holds true for the object's marginal, that is Equation 5.3.5 summation would be over the agents variable . Figure 5.6 illustrates the joint distribution of both the agent and the object random variable. For ease of notation we use the shorthand $P(A_t)$ for $P(A_t|Y_{0:t}, u_{1:t})$. The 1D world of the agent and object is discretised to 10 states, producing a joint distribution with 100 parameters! For a state space of N bins, $i = 1...N$, and there is a total of M random variables (one agent and $M - 1$ objects) and the joint distribution has N^M parameters. This exponential increase renders Histogram-SLAM intractable with this parameterisation.

Histogram likelihood model

A measurement model, $P(Y_t|A_t, O)$, predicts the likelihood of observations, Y_t, given the state of A_t and O. In the tasks we consider, an observation occurs only if the agent enters in contact with the object, which implies that both occupy the same discrete state.

$$P(Y_t = 1|A_t, O) = \begin{cases} 1 & \text{if } A_t = O \\ 0 & \text{if } A_t \neq O \end{cases} \tag{5.3.7}$$

Figure 5.7, illustrates the likelihood of Equation 5.3.7 in the case when a no contact measurement $Y_t = 0$ is present in a 1D world. Both likelihoods are sparse in the sense that there is only a small region which gets affected by the likelihood. When there is no measurement (*Left*) all the parameters of the joint

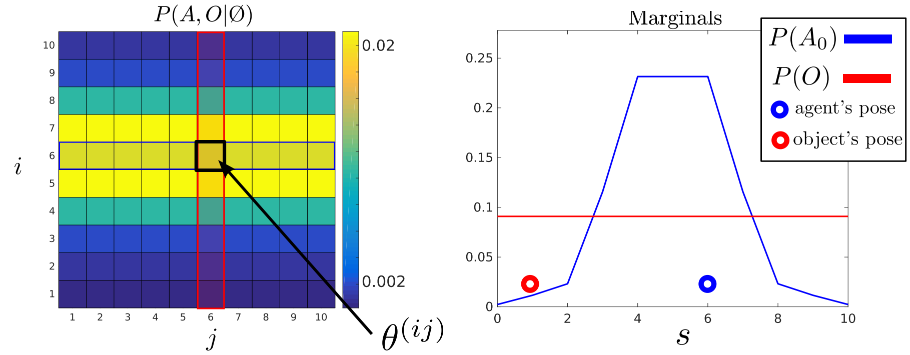

Figure 5.6: *Left:* Initialisation of the agent and object joint distribution. *Right:* Marginals of the agent and object, giving the probability of their location. The marginal of each random variable is obtained from Equation 5.3.5. The probability of the agent and object being in state $s = 6$ is given by summing the blue and red highlighted parameters in the joint distribution.

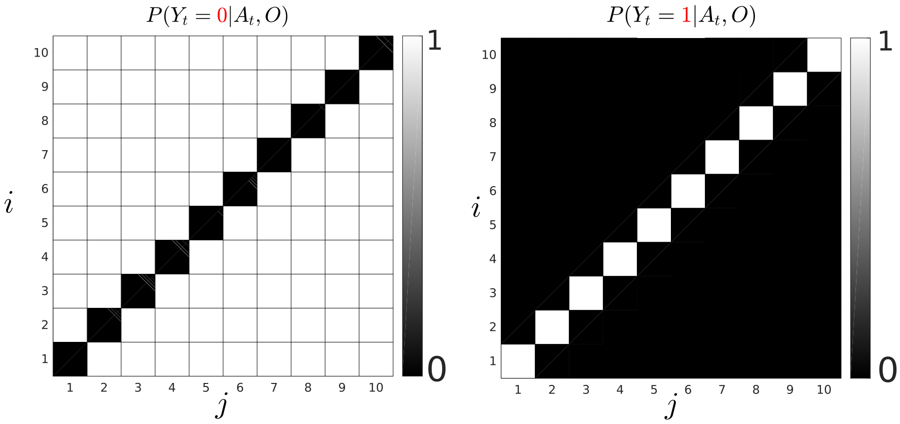

Figure 5.7: 1D world Likelihood $P(Y_t|A_t, O)$. *Left:* No contact detected with the object, the current measurement is $Y_t = 0$, both the agent and object cannot be in the same state. *Right:* The agent entered into contact with the object and received a haptic feedback $Y_t = 1$. There are only two measurements contact or no contact, which the agent receives.

distribution which are in the black regions will become zero. When the object is detected (*Right*) then only parameters of the joint distribution which are within the white regions will be non-zero. All changes to the joint distribution are constrained to the states $i = j$, which in the case of a 2-dimensional joint distribution is a line. The **sparsity** of the likelihood function will be key to the development of the MLMF filter.

Two models are needed to perform the recursion, namely the motion model $P(A_t|A_{t-1}, u_t)$ and the measurement model $P(Y_t|A_t, O)$, which we already detailed. Both models applied consecutively to the initial joint distribution results in a posterior distribution. Both Equation 5.3.9-5.3.10 are part of the histogram Bayesian filter update:

Discrete Bayesian filter recursion

intialisation

$$P(A_0, O) = P(A_0)P(O) = \boldsymbol{\theta}^{(i)}\boldsymbol{\theta}^{(j)} = \boldsymbol{\theta}^{(i,j)} \tag{5.3.8}$$

motion

$$P(A_t, O|Y_{0:t-1}, u_{1:t}) = \sum_{A_{t-1}} P(A_t|A_{t-1}, u_t)P(A_{t-1}, O_t|Y_{0:t-1}, u_{1:t-1}) \tag{5.3.9}$$

measurement

$$P(A_t, O|Y_{0:t}, u_{1:t}) = \frac{P(Y_t|A_t, O)P(A_t, O|Y_{0:t-1}, u_{1:t})}{P(Y_t|Y_{0:t-1})} \tag{5.3.10}$$

For the derivation of these two steps, the reader is referred to Appendix 5.7.2. Figure 5.8 illustrates how the joint distribution evolves in a 1D world. The agent and object's true positions (unobservable) are in state 6 and 1. The agent moves four steps towards state 10. At each time step, as the agent does not sense the object, the likelihood function $P(Y_t = 0|A_t, O)$ (Figure 5.7 *Left*) is applied. As the agent moves towards the right, the motion model shifts the joint distribution towards state 10 along the agent's dimension, i (note that state 1 and 10 are wrapped).

As the agent moves to the right more joint distribution parameters become zero and the re-normalisation by the **evidence** ($P(Y_t|Y_{0:t-1}$, denominator of Equation 5.3.10), which increases the value of the remaining parameters, is equal to the sum of the probability mass which was set to zero by the likelihood function. Thus the values of the parameters of the joint distribution which fall on the pink line in Figure 5.8 (green line also, but only for first time slice) become zero and their values are redistributed to the remaining non-zero parameters. This is an **important aspect** which will be present in MLMF and we define this to be $\alpha \in \mathbb{R}$.

When the agent enters into contact with the object and senses it, the likelihood function $P(Y_t = 1|A_t, O)$, Figure 5.7 (*Right*), is multiplied with the joint distribution with the result illustrated in Figure 5.9. Only parameters of the joint distribution whose indices satisfy $i = j$ will remain unchanged, all the others will become zero. In retrospect the likelihood $P(Y_t = 1|A_t, O)$ acts as a **constraint**, that is, the agent and object have to be in the same state given by a line traversing the 2D joint distribution.

The **inconvenience** with Histogram-SLAM is that its time and space complexity is exponential, as the joint distribution is discretized and parametrised by $\boldsymbol{\theta}^{(ij)}$. Instead we propose a filter, named MLMF, which does not parameterise the joint distribution.

The particularity of the new filter, MLMF, which we formally introduce in the next section, is to achieve the same result as the Histogram filter but without having to parameterise the joint distribution, thus avoiding the exponential

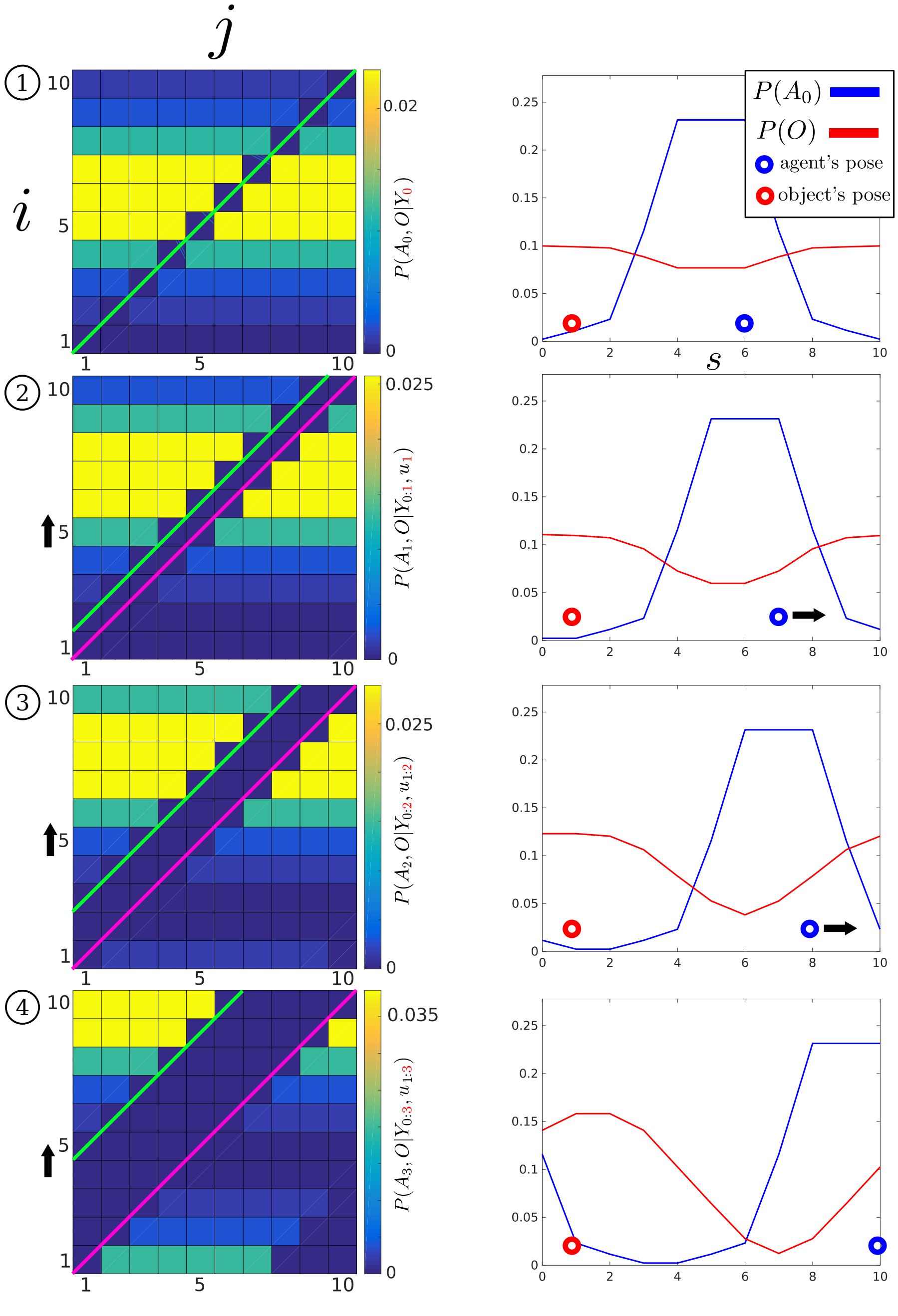

Figure 5.8: Histogram-SLAM, 4 time steps. **1** Application of likelihood $P(Y_0 = 0|A_0, O)$ and the agent remains stationary, $u_0 = 0$, all states along the green line become zero. **2** The agent moves to the right $u_1 = 1$, the motion $P(A_1|A_0, u_1)$, and likelihood models are applied consecutively. The right motion results in a shift (black arrow on the left) in the joint probability distribution towards the state $i = 10$. All parameters on the pink line are zero. **3** Same as two. **4** The original result of the likelihood function, green line, has moved by the same amount as the agent's displacement. At each time step a new likelihood function (pink line) is applied to the joint distribution.

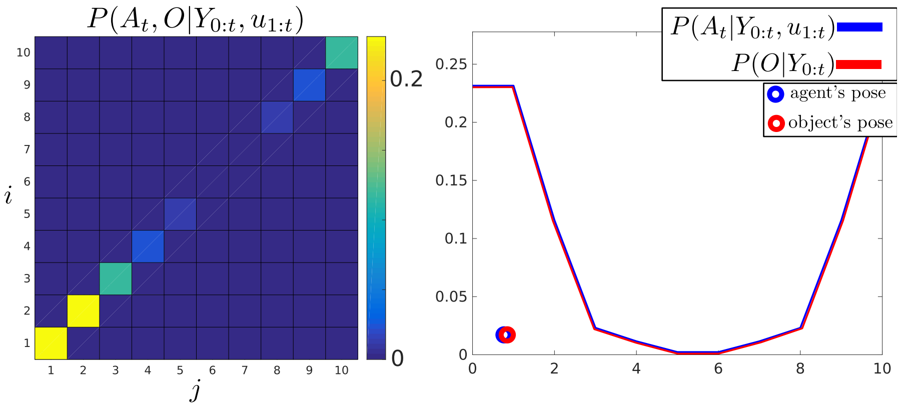

Figure 5.9: Histogram-SLAM contact. The agent has entered in contact with the object (measurement $Y_t = 1$) and the likelihood function $P(Y_t = 1|A_t, O)$ is applied to the joint distribution. Only parameters on the line $i = j$ will remain unchanged and parameters for which $i \neq j$ will be set to zero.

growth cost. The **key idea** behind the mechanism of the MLMF filter is to only evaluate the joint distribution in states (i, j) for which the likelihood is zero and apply the updates directly to the marginals without parameterising the values of the joint distribution. The MLMF filter parametrises **explicitly** the marginals $P(A_t = s|Y_{0:t}, u_{1:t}) : \boldsymbol{\theta}_a^{(s)}$, $P(O = s|Y_{0:t}, u_{1:t}) : \boldsymbol{\theta}_o^{(s)}$. This is in contrast to the Histogram filter in which the marginals are derived from the joint distribution by marginalisation. For the MLMF filter, in order to respect the Bayesian recursion, the filter has to **memorise** the complete history of likelihood functions $\{P(Y_t|A_t, O)\}_{t=0:T}$ and the normalisation coefficient α. The reasons for this will be made clear in the next section. Below we summarise the parameters of the Histogram and the desired MLMF parameters:

Histogram	:	$\boldsymbol{\theta}^{(ij)}$
desired-MLMF	:	$\boldsymbol{\theta}_a^{(s)}, \boldsymbol{\theta}_o^{(s)}, \alpha, \{(Y, l)_p\}_{p=0:t}$
		$s, i, j = 1, \ldots, N$

The likelihood function is parametrised by a measurement, Y_t and offset l to the i-axis. In Figure 5.8 we see that the likelihood applied at the first time step (green line) is superimposed on the line $i = j$. As actions are applied, $u_{1:3} = [1, 1, 1, 1]$, the first likelihood function (green line) shifts by an amount corresponding to the total displacement (4 in this case), see the last sub-figure of 5.8. The second likelihood applied at $t = 1$ will have an offset corresponding to the sum $u_{1:3}$, the third will have an offset of $u_{2:3}$ and so on and so forth.

The MLMF, which we mathematically derive in the next section, keeps as parameters the marginals, likelihood function parameters and a normalisation constant. The marginals are updated (filter recursion) by **evaluating** the joint distribution in states which are set to zero by the current applied likelihood and removing the value of the zeroed parameters from the marginals.

When a positive contact is established with the object, the likelihood is zero everywhere other than the states corresponding to $i = j$, see Figure 5.9. In this case there is no need to evaluate the likelihood elsewhere in the joint distribution. The marginals are simply the normalisation of the line $i = j$. The evaluation of the joint distribution, given that the object is sensed or not, is restricted to the states (i, j) for which $i = j$ and there are a total of N states. This is much less than N^2 which Histogram-SLAM requires.

5.4 Measurement Likelihood Memory Filter

To obtain an informative update of the marginals, the random variables A and O must remain dependent in the absence of a positive observation Y. Additionally, this dependence should be efficiently encoded in contrast to the histogram parametrisation shown in Section 5.3.

The likelihood function $P(Y_t|A_t, O)$ is the cause of the dependence between the random variables. If both the agent and object were completely independent, no additional parameters would be required to represent the joint distribution other than the marginals $P(A_t)$ and $P(O)$ giving a total of NM parameters (where N is the number of states and M is the number of random variables). At the other extreme if every single point in the domain of the random variables was dependent this would require the totality of N^M parameters as previously stated in the case of the histogram filter. We propose a method in which we do not model the joint distribution explicitly but rather only compute its impact on the marginals.

5.4.1 MLMF parametrisation

The concept behind MLMF is to keep a **function parameterisation** of the joint distribution instead of a **value parameterisation** as it is the case for Histogram-SLAM. At initialisation the joint distribution is represented by the product of marginals, Equation 5.4.1, which would result in the joint distribution illustrated in Figure 5.6, if it were to be evaluated at all states (i, j) as it is done for Histogram-SLAM, Equation 5.3.8. MLMF will only evaluate this product at specific states, only when necessary. At each time step the motion and measurement update are applied, Equation 5.4.2-5.4.2. An important distinction is that these updates are performed on the joint distribution, which is not the case in Histogram-SLAM where the updates are done on the conditional, Equation 5.3.9-5.3.10. After applying multiple motion and measurement updates the resulting joint distribution is given by Equation 5.4.4, see Appendix 5.7.3 for a step-by-step derivation. The filtered condition, Equation 5.4.5, is the result of the normalisation of the filtered joint by the marginal likelihood.

MLMF Bayesian filter

intialisation

$$P(A_0, O) = P(A_0)P(O) \tag{5.4.1}$$

motion

$$P(A_t, O, Y_{0:t-1}|u_{1:t}) = \sum_{A_{t-1}} P(A_t|A_{t-1}, u_t)P(A_{t-1}, O, Y_{0:t-1}|u_{1:t-1}) \tag{5.4.2}$$

measurement

$$P(A_t, O, Y_{0:t}|u_{1:t}) = P(Y_t|A_t, O)P(A_t, O, Y_{0:t-1}|u_{1:t-1}) \tag{5.4.3}$$

filter (joint)

$$P(A_t, O, Y_{0:t}|u_{1:t}) = P(O)P(A_t|u_{1:t}) \prod_{i=0}^{t} P(Y_i|A_t, O, u_{i:t}) \tag{5.4.4}$$

filter (conditional)

$$P(A_t, O|Y_{0:t}, u_{1:t}) = \frac{P(A_t, O, Y_{0:t}|u_{1:t})}{P(Y_{0:t}|u_{1:t})} \tag{5.4.5}$$

The motion update, Equation 5.4.2, when applied to the joint distribution results in the initial marginal $P(A_0)$ and the likelihood functions being moved along the agent's axis. Equation 5.4.5, would require an expensive evaluation of Equation 5.4.4 at each state (i, j) resulting in the evaluation of the likelihood product. We will demonstrate that MLMF only evaluates dependent states (i, j) (states which are affected by the likelihood). The MLFM filter is parameterised by the agent and object marginals $P(A_t|u_{1:t})$, $P(O)$ and the history of likelihood functions, Equation 5.4.4, which is all the likelihood functions since $t = 0$ until t and we will refer to it as the **memory likelihood function**:

$$P(Y_{0:t}|A_t, O, u_{0:t}; \Psi_{0:t}) := \prod_{i=0}^{t} P(Y_i|A_t, O, u_{i:t}) \tag{5.4.6}$$

$$P(Y_t = 0|A_t, O, u_{i:t} = l \begin{cases} 0 & \text{if } A_t = O \\ 1 & \text{else} \end{cases} \tag{5.4.7}$$

The likelihood functions have two different functional forms (see Figure 5.7) which is dictated by the current binary measurement and the offset of the likelihood function along the agent's axis. The parameters of Equation 5.4.6 are all the measurements since the start of the filtering and offsets along the agent's axis, we refer to these parameters as $\Psi_{0:t} = \{(Y, l)_p\}_{p=0:t}$. In Algorithm 2, we detail how an action and measurement at time step t, result in the update of $\Psi_{0:t}$, the parameters of the likelihood functions; this is an implementation of Equation 5.4.2 for the likelihood product.

Algorithm 2: Memory Likelihood update

```
input  : $\Psi_{0:t-1}$, $Y_t$, $u_t$
output: $\Psi_{0:t}$

motion update $\bar{\Psi}_{0:t} \leftarrow \Psi_{0:t-1}$
for $l \in \Psi_{0:t-1}$ do
    $l = l + u_t$

measurement update
$\Psi_{0:t} \leftarrow \{\bar{\Psi}_{0:t}, (Y_t, l = 0)\}$
```

Figure 5.10 illustrates the evolution of the MLMF joint distribution, Equation 5.4.4. Both $P(A_0)$ and $P(O)$ where initialised the same as for the Histogram-SLAM example in Figure 5.8 on page 129. Two actions $u_{1:2} = 1$ are applied and three measurements $Y_{0:3} = 0$ received which are then integrated into the filter. Since initialisation of the joint distribution at $t = 0$ until $t = 3$ the object's marginal $P(O)$ remains unchanged and the agent's marginal $P(A_2|u_{1:2})$ has been updated by the two actions according to the motion update, see Figure 5.10 *Top-right*. The product of these two marginals (terms of Equation 5.4.4 before the memory likelihood product) result in the joint $P(A_2, O|u_{1:2})$ illustrated in Figure 5.10 *Middle-right*.

In the left column of Figure 5.10 we illustrate how the memory likelihood term, Equation 5.4.6, gets updated according to Algorithm 2. In the *Top-left*, the first likelihood function $P(Y_0|A_2, O, u_{1:2})$ is illustrated. As two actions have been applied since it was added to the memory term, the motion update of Algorithm 2 has been applied twice which results in a $l = 2$ parameter. We highlighted the likelihood in green to indicate that it was the first added to the memory term and for easy comparison with Figure 5.8 on page 129. As for the second likelihood function $P(Y_1|A_2, O, u_2)$ only one action has been applied to it and the third likelihood $P(Y_3|A_2, O)$ has not been updated by the next action u_3. The parameters of likelihood product, Equation 5.4.6, are: $\Psi_{0:2} = \{(0, 2)_{p=0}, (0, 1)_{p=1}, (0, 0)_{p=2}\}$ and the evaluation of memory likelihood is depicted in the *Bottom-left* of Figure 5.10.

The reader may have noticed that the amplitude of the values of the filtered joint distribution illustrated in Figure 5.10 have not changed when compared with Figure 5.8. This is because we have not re-normalised the joint distribution by the evidence $P(Y_{0:t}|u_{1:t})$. We will show in the next section how we can **recursively** compute the evidence without having to integrate the whole joint distribution which would be very expensive.

For ease of notation from this point onwards we will not show the conditioned actions $u_{1:t}$, so $P(A_t, O|Y_{0:t}, u_{1:t})$ will be $P(A_t, O|Y_{0:t})$.

Our goal is to be able to compute the marginals $P(A_t|Y_{0:t})$, $P(O|Y_{0:t})$ of the agent and object random variables and marginal likelihood $P(Y_{0:t}|u_{1:t})$ **without**

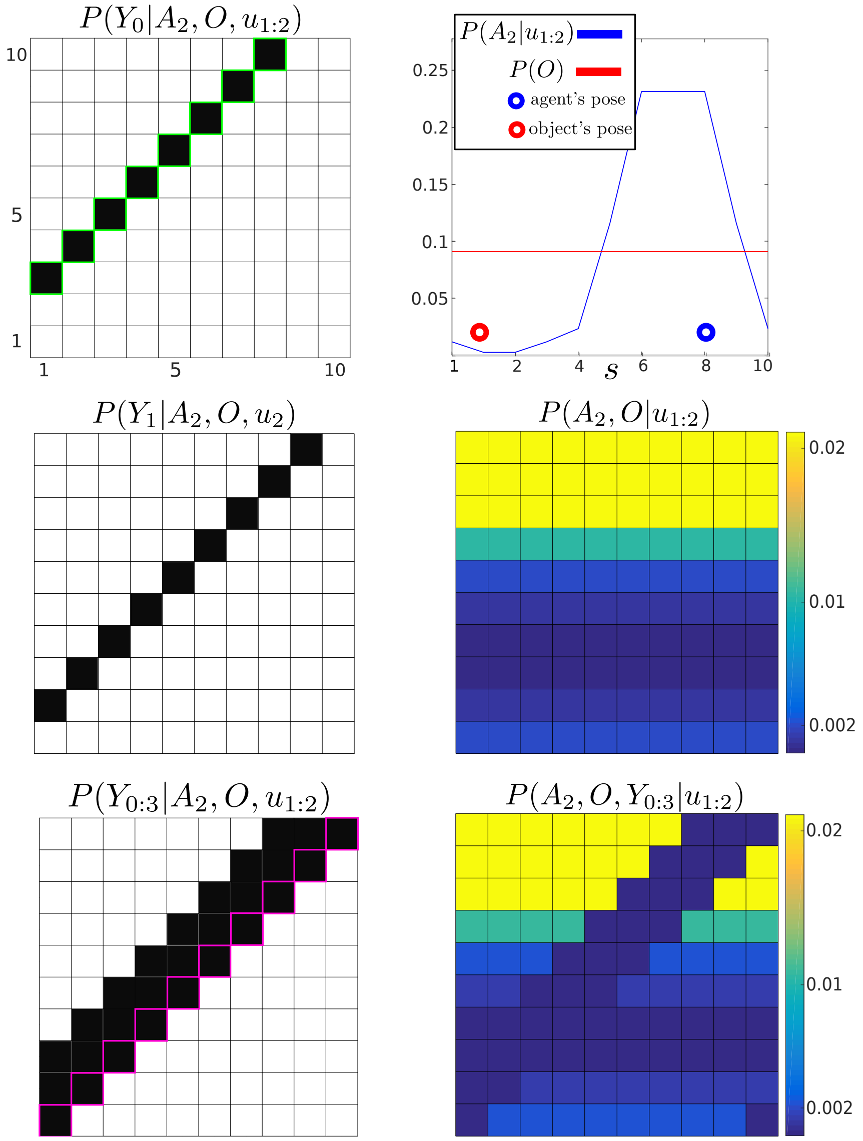

Figure 5.10: The MLMF joint distribution, Equation 5.4.4, at time $t = 3$. Three measurements (all $Y = 0$) and two actions (both $u = 1$) have been integrated into the joint distribution. *Left column:* The first plot illustrates the likelihood of the first measurement Y_0, we highlight it in green to indicate it was the first applied likelihood function (see corresponds with Figure 5.8). The first likelihood function has been moved by the 2 actions, the second likelihood function has been moved by one action (the last one, $u_2 = 1$) and the third likelihood has not has had no action applied to it yet. The last applied likelihood function is highlighted in pink and product of all the likelihoods since $t = 0$ until $t = 3$ is depicted at the bottom of the figure which is $P(Y_{0:3}|A_2, O, u_{1:2})$. *Right column:* the top figure illustrates the original marginal of the object $P(O)$, which remains unchanged, and the agent's marginal $P(A_2|u_{1:2})$ which has moved in accordance to the motion update function. Their product would results in the joint distribution $P(A_2, O|u_{1:2})$ illustrated in the middle figure if evaluated at each state (i, j). The bottom figure is the result of multiplying $P(A_2, O|u_{1:2})$ with $P(Y_{0:3}|A_2, O, u_{1:2})$ gives the filtered joint distribution, Equation 5.4.4.

having to perform an **expensive marginalisation** over the entire space of the joint distribution as was the case for Histogram-SLAM.

The next section describes how to efficiently compute the normalisation factor, denominator Equation 5.4.5, and the marginals.

5.4.2 Computation of evidence and marginals

In order to compute efficiently the marginal likelihood (also known as evidence) $P(Y_{0:t}|u_{1:t})$ and the filtered marginals $P(A_t|Y_{0:t})$, $P(O|Y_{0:t})$ we take advantage of the fact that only a very small area in the joint distribution space will be affected by the measurement likelihood function at each time step.

Without lost of generality the likelihood function will only make a difference to dependent $A \cap O$ regions in the joint distribution, areas where the likelihood function is less than one. The region inside $A \ominus O$ will not be affected, where the likelihood function is equal to one. Figure 5.11 shows the relation between the measurement function $P(Y_t|A_t, O)$ and the joint distribution $P(A_t, O|Y_{0:t})$ for three different initialisations.

As illustrated and explained in Figure 5.11, the joint distribution can be decomposed in a dependent and independent term (Equation 5.4.8).

$$P(A_t, O|Y_{0:t}) = P_{\cap}(A_t, O|Y_{0:t}) + P_{\ominus}(A_t, O|Y_{0:t}) \tag{5.4.8}$$

The probability mass covered by the dependent term is located within the measurement function's tube and the independent probability mass is located outside as shown in Figure 5.11. This formulation will lead to large computational gain as the independent term is not influenced by the measurement function: $P_{\ominus}(A_t, O, \mathbf{Y_{0:t}}) = P_{\ominus}(A_t, O, \mathbf{Y_{0:t-1}})$ and $P_{\ominus}(A_t, O|\mathbf{Y_{0:t}}) \propto P_{\ominus}(A_t, O|\mathbf{Y_{0:t-1}})$.

Evidence

The evidence of the measurement $P(Y_{0:t}|u_{1:t})$ is the normalisation coefficient of the joint distribution Equation 5.4.5. It is the amount of probability mass re-normalised to the other parameters as a result of the consecutive application of the likelihood function. At time step t, the normalising factor to be added to the evidence is the difference between the probability mass located inside $A \cap O$ before and after the application of the measurement function $P(Y_t|A_t, O)$, see Equation 5.4.9-5.4.10 (see Appendix 5.7.4 for the full derivation).

$$\alpha_t = \sum_{A_t} \sum_{O} \Big(P(Y_t|A_t, O) - 1 \Big) P_{\cap}(A_t, O, Y_{0:t-1}|u_{1:t}) \tag{5.4.9}$$

$$P(Y_{0:t}|u_{1:t}; \alpha_{0:t}) = 1 + \underbrace{\alpha_{0:t-1} + \alpha_t}_{\alpha_{0:t}} \tag{5.4.10}$$

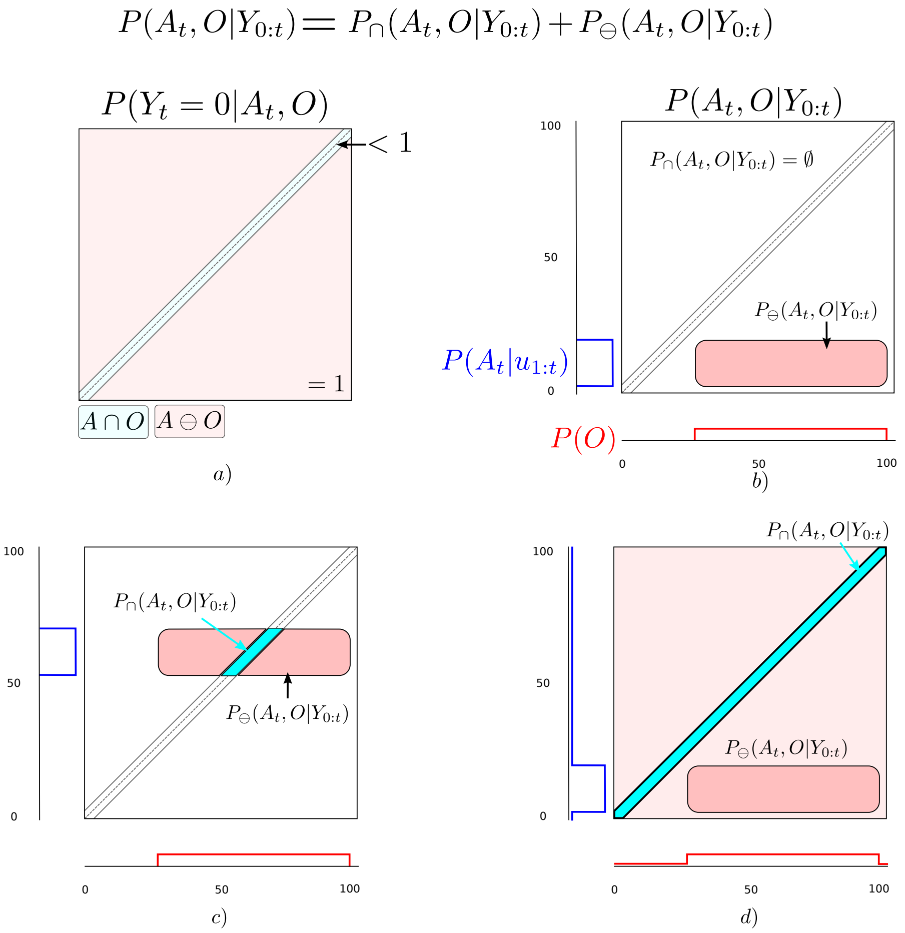
$P(A_t,O|Y_{0:t}) = P_{\cap}(A_t,O|Y_{0:t}) + P_{\ominus}(A_t,O|Y_{0:t})$
$P(Y_t = 0|A_t,O)$
< 1
$= 1$
$A \cap O$
$A \ominus O$
$a)$
$P(A_t,O|Y_{0:t})$
100
50
0
$P_{\cap}(A_t,O|Y_{0:t}) = \emptyset$
$P_{\ominus}(A_t,O|Y_{0:t})$
$P(A_t|u_{1:t})$
$P(O)$
0
50
100
$b)$
100
50
0
$P_{\cap}(A_t,O|Y_{0:t})$
$P_{\ominus}(A_t,O|Y_{0:t})$
0
50
100
$c)$
$P_{\cap}(A_t,O|Y_{0:t})$
100
50
0
$P_{\ominus}(A_t,O|Y_{0:t})$
0
50
100
$d)$

Figure 5.11: **a)** Likelihood $P(Y_t = 0|A_t, O)$, the blue area depicts the regions in which the likelihood is < 1 and the red area is where the likelihood is $= 1$. If the probability mass of the joint distribution is in the blue region, then the parameters of the random variables in these areas are dependent, $A \cap O$ and otherwise they are independent from one another, $A \ominus O$. **b)** The agent and object marginals are not overlapping and thus are completely independent. The joint distribution, $P(A_t, O|Y_{0:t})$ the black rectangle, is not intersecting with the measurement function. As a result $P_{\cap}(A_t, O|Y_{0:t})$ is empty. **c)** The marginals overlap resulting in the measurement likelihood function intersecting with the joint distribution. The joint distribution is composed of the blue and red areas, Equation 5.4.8. The probability mass at the intersection gets removed and re-normalised to other regions which is the result of applying Bayes integration. **d)** The marginals of A and O are completely overlapping, however only a small fraction of the probability mass in the joint distribution is within the measurement function's tube.

The advantage of Equation 5.4.9 is that the summation is only over states which are in the dependent area $\cap$ of the joint distribution. This is generally always much smaller than the full space itself. Until an object is sensed, the likelihood will always be zero $P(Y_t|A_t, O) = 0$ and α_t will correspond to the probability mass which falls within the region of the joint distribution in which the likelihood function is zero. In Figure 5.11 b) & d), the sum of the probability mass in the blue regions is equal to α_t. The point of interest is that as we perform the filtering process we will never re-normalise the whole joint distribution, but only keep track of how much it should have been normalised. To this end the marginals $P(A_t|u_{1:t})$ and $P(O)$ are never re-normalised but are used to compute at each step how much of the probability mass α_t should go to the normalisation factor $P(Y_{0:t}|u_{1:t}; \alpha_{0:t})$. The evidence in question will never be negative, as the joint distribution sums to one and each α_t represents some of the mass removed from the joint distribution. Since we keep track of the history of applied measurement likelihood functions we never remove the same amount of probability mass twice from the joint distribution.

Marginals

There are two different types of marginals used in the MLMF filter. The first set are the initial **joint marginals** of the joint distribution, Equation 5.4.4. The second set of marginals are the **filtered marginals** which are updated by evaluating the joint distribution in dependent states.

Marginals

joint marignals
$P(A_t|u_{1:t})$ and $P(O)$: Equation 5.4.4
filtered marignals
$P(A_t|Y_{0:t}, u_{1:t})$ and $P(O|Y_{0:t})$

In Histogram-SLAM both the agent and object marginals are obtained, at every time, by marginalising the joint distribution. This requires storing and summing over all the parameters of the joint distribution which is expensive. Instead in MLMF we take advantage of the sparsity of the likelihood function and as a result we take only the dependent marginal components into consideration, Equation 5.4.11.

$$P(A_t|Y_{0:t}) = P(A_t|Y_{0:t-1}) - \Big(P_\cap(A_t|Y_{0:t-1}) - \mathbf{P}_\cap(\mathbf{A_t}|\mathbf{Y_{0:t}})\Big) \tag{5.4.11}$$

(see Appendix 5.7.5 for the full derivation of Equation 5.4.11).

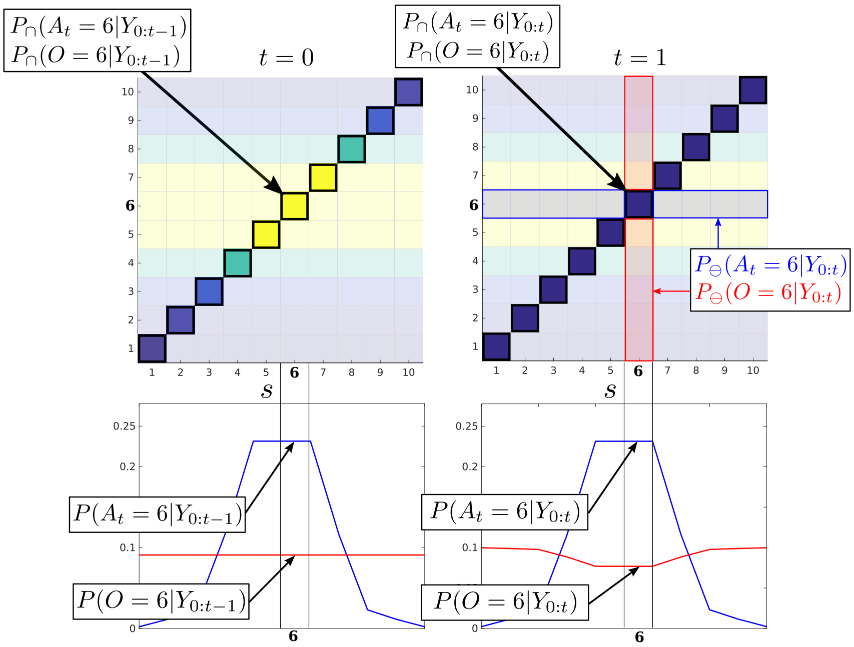

Figure 5.12: Filtered Marginals. Illustration of the agent and object marginal update, Equation 5.4.11. The parameters of the joint distribution which are independent $A \ominus O$ are pale and the dependent areas $A \cap O$, where $P(Y_t < 1|A_t, O)$, are bright. MLMF only evaluates the joint distribution in dependent states. For each state s of the marginals $1, \ldots, 10$ the difference of the marginals inside the dependent area, before and after the measurement likelihood is applied, is evaluated and removed from the marginals $P(A_t|Y_{0:t-1})$, $P(O|Y_{0:t-1})$ leading to $P(A_t|Y_{0:t})$, $P(O|Y_{0:t})$ (we did not show $u_{1:t}$ for easing the notation).

$$\mathbf{P}_\cap(\mathbf{A_t}|\mathbf{Y_{0:t}}) = \frac{\sum_O \overbrace{P(Y_t|A_t,O)P_\cap(A_t,O,Y_{0:t-1}|u_{1:t};\bar{\Psi}_{0:t})}^{P_\cap(A_t,O,Y_{0:t}|u_{1:t})}}{\underbrace{1+\alpha_{0:t}}_{P(Y_{0:t}|u_{1:t};\alpha_{0:t})}} \tag{5.4.12}$$

$$= \sum_O P_\cap(A_t, O|Y_{0:t}) \tag{5.4.13}$$

Equation 5.4.11 is recursive $P(A_t,|Y_{\mathbf{0:t}})$, it is computed in terms of $P(A_t|Y_{\mathbf{0:t-1}})$. In Figure 5.12 we illustrate how the MLMF updates the marginals of the agent and object given a new measurement. The illustrated marginals (*Bottom row*) are the **filtered marginals** $P(A_t|Y_{0:t}, u_{1:t})$, $P(O|Y_{0:t})$. The shape of the **joint marginals** $P(A_t|u_{1:t})$, $P(O)$ (not shown here, see Figure 5.10) will remain unchanged by measurements during the filtering process. They are the parameters of the joint distribution used to update the filtered marginals.

Table 5.1 summarises the functions and parameters of the MLMF for two random variables, an agent and object.

functions		parameters	description
$P(A_t\|Y_{0:t}, u_{1:t})$	:	$\boldsymbol{\theta}_a$	filtered marginals
$P(O\|Y_{0:t})$	:	$\boldsymbol{\theta}_o$	
$P(A_t\|u_{1:t})$	:	$\boldsymbol{\theta}_a^*$	joint marginals
$P(O)$	:	$\boldsymbol{\theta}_o^*$	
$P(Y_{0:t}\|u_{1:t})$	:	$\alpha_{0:t} \in \mathbb{R}$	evidence
$P(Y_{0:t}\|A_t, O, u_{1:t})$	:	$\Psi_{0:t} = \{(Y, l)_p\}_{p=0:t}$	likelihood history

Table 5.1: MLMF functions with associated parameters. The marginal parameters are the discretisation of the state space $\boldsymbol{\theta} \in \mathbb{R}^N$, $\boldsymbol{\theta}^{(s)}$ corresponds to the probability being in state s.

5.4.3 MLMF-SLAM Algorithm

In Algorithm 3 we detail the motion-measurement update steps of MLMF-SLAM. This formulation is advantageous as the joint distribution is only evaluated inside the dependent regions $A \cap O$ of the joint distribution. All the **parameters** $\Psi_{0:t}$ of the memory likelihood function $P(Y_{0:t}|A_t, O, u_{1:t}; \Psi_{0:t})$ are retained. We keep track of the normalisation factor, the evidence $P(Y_{0:t}|u_{1:t}; \alpha_{0:t})$ which is a scalar quantity. With these parameters it is possible to reconstruct the joint distribution, however the MLMF-SLAM algorithm only makes changes to the filter marginals, see Table 5.1.

We evaluated this formulation of the joint distribution with the standard histogram filter in the case of the 1D filtering scenario illustrated in Figure 5.8 and we found them to be identical. Having respected the formulation of Bayes rule, we assert that Algorithm 3 is a Bayesian Optimal Filter[1].

We next evaluate both space and time complexity of the MLMF filter.

[1] An optimal Bayesian solution is an exact solution to the recursive problem of calculating the exact posterior density Arulampalam et al. (2002b)

```
Algorithm 3: MLMF-SLAM
input :
        measurements
        $\mathbf{Y_t}$, $\mathbf{u_t}$
        joint distribution parameters:
        $P(A_{t-1}|u_{1:t-1})$ $P(O)$, $\Psi_{0:t-1}$, $\alpha_{0:t-1}$
        filtered marginals:
        $P(A_{t-1}|Y_{0:t-1}, u_{1:t-1})$, $P(O|Y_{0:t-1})$
output:
        joint parameters:
        $P(A_t|u_{1:t})$, $\Psi_{0:t}$, $\alpha_{0:t}$
        filtered marginals:
        $P(A_t|Y_{0:t}, u_{1:t})$, $P(O|Y_{0:t})$

motion update
    $P(A_t|u_{1:t}) = \sum_{A_{t-1}} P(A_t|A_{t-1}, \mathbf{u_t}) P(A_{t-1}|u_{1:t-1})$
    $P(A_t|Y_{0:t-1}, u_{1:t}) = \sum_{A_{t-1}} P(A_t|A_{t-1}, \mathbf{u_t}) P(A_{t-1}|Y_{0:t-1}, u_{1:t-1})$
    $\bar{\Psi}_{0:t} \leftarrow \Psi_{0:t-1}$ : Algorithm 2 (motion update)

measurement update
    $\alpha_{0:t} = \alpha_{0:t-1} + \sum_{A_t} \sum_{O} \Big( P(\mathbf{Y_t}|A_t, O) - 1 \Big) P_{\cap}(A_t, O, Y_{0:t-1}|u_{1:t})$
    $P(Y_{0:t}|u_{1:t}) = 1 + \alpha_{0:t}$
    $P(A_t|Y_{0:t}) = P(A_t|Y_{0:t-1}) - \Big( P_{\cap}(A_t|Y_{0:t-1}) - P_{\cap}(A_t|Y_{0:\mathbf{t}}) \Big)$
    $P(O|Y_{0:t}) = P(O|Y_{0:t-1}) - \Big( P_{\cap}(O_t|Y_{0:t-1}) - P_{\cap}(O_t|Y_{0:\mathbf{t}}) \Big)$
    $\Psi_{0:t} \leftarrow \bar{\Psi}_{0:t}$ : Algorithm 2 (measurement update)
```

5.4.4 Space & time complexity

For discussion purposes we consider the case of three beliefs, namely that of the agent and two other objects $O^{(1)}$ and $O^{(2)}$ which we subsequently generalise. As stated previously M stands for the number of filtered random variables including the agent. N is the number of discrete states in the world. In the following section, we compare the space and time complexity of MLMF-SLAM with Histogram-SLAM.

Space complexity

Figure 5.13 *left* illustrates the volume occupied by the joint distribution for a space with N states. Histogram-SLAM would require N^3 parameters for the joint distribution $P(A, O^{(1)}, O^{(2)})$ and N^M for M random variables.

For MLMF-SLAM, each random variable requires two sets parameters $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^*$ (see Table 5.1). Given M random variables, the initial number of parameters is $M(2N)$. At every time step the likelihood memory function increments by one measurement and offset, $(Y_t, l = 0)$ (Algorithm 2). Given a state space of size N, there can be no more than N different measurement functions (one for each state). In the worst case scenario the number of parameters $\Psi_{0:t}$ of the memory likelihood function, Equation 5.4.6, will be N. The total number of parameters is $M(2N) + N$ which gives a final worst case space complexity which is linear in the number of random variables, $\mathcal{O}(N\,M)$.

Time complexity

For Histogram-SLAM, the computational cost is equivalent to that of the space complexity, $\mathcal{O}(N^M)$, since every state in the joint distribution has to be summed to obtain all the marginals.

For MLMF-SLAM, every state in the joint distribution's state space which has been changed by the measurement likelihood function has to be summed, see Figure 5.12 on page 138. As a result the computational complexity is directly related to the number of dependent states $|A \cap O|$. In Figure 5.12, this corresponds to states where $i = j$ and there are N out of a total N^2 states in the joint distribution. In Figure 5.13 *left* we illustrate a joint distribution with N^3 states and the dependent states $|A \cap O^{(1)} \cap O^{(2)}|$ are those which are within the blue and red planes (where the likelihood evaluates to zero). The blue and red plain comprise of N^2 states each giving a total of $2\,N^2$ dependent states. Lets see why it is N^2 for each likelihood, the overall likelihood is given by Equation 5.4.14:

$$P(Y_t|A_t, O^{(1)}, O^{(2)}) = P(Y_t|A_t, O^{(1)})\,P(Y_t|A_t, O^{(2)}) \tag{5.4.14}$$

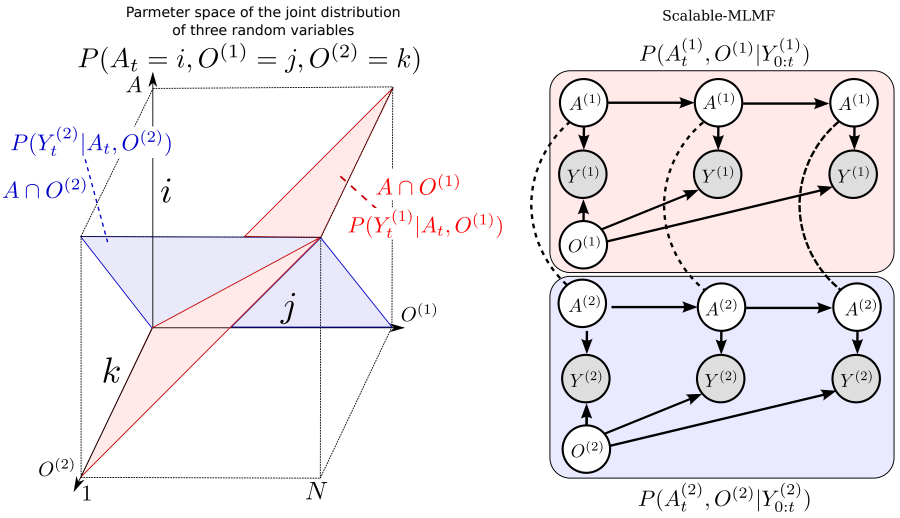
Parmeter space of the joint distribution
of three random variables
$P(A_t = i, O^{(1)} = j, O^{(2)} = k)$
A
$P(Y_t^{(2)}|A_t, O^{(2)})$
$A \cap O^{(2)}$
i
$A \cap O^{(1)}$
$P(Y_t^{(1)}|A_t, O^{(1)})$
j
$O^{(1)}$
k
$O^{(2)}$
1
N
Scalable-MLMF
$P(A_t^{(1)}, O^{(1)}|Y_{0:t}^{(1)})$
$A^{(1)}$
$Y^{(1)}$
$O^{(1)}$
$A^{(2)}$
$Y^{(2)}$
$O^{(2)}$
$P(A_t^{(2)}, O^{(2)}|Y_{0:t}^{(2)})$

Figure 5.13: *Left:* Joint distribution $P(A, O^{(1)}, O^{(2)})$ of the agent and two objects. Each measurement likelihood function, $P(Y|A, O^{(1)})$, $P(Y|A, O^{(2)})$ corresponds to a hyperplane in the joint distribution The state space is discretised to N bins giving the total number of parameters in the joint distribution of N^3. *Right:* **Scalable-MLMF** Each agent-object joint distribution pair is modelled independently. For clarity we have left out the action random variable u which is linked to every agent node. Two joint distributions $P(A^{(1)}, O^{(1)}|Y_{0:t}^{(1)})$ and $P(A^{(2)}, O^{(2)}|Y_{0:t}^{(2)})$ parametrise the graphical model. The dashed undirected lines represent a wanted dependency, if present $O^{(1)}$ and $O^{(2)}$ are to be dependent through A. In the standard setting there will be no exchange of information between the individual joints. However we demonstrate later on how we perform a one time transfer of information when one of the objects is sensed.

The likelihood term $P(Y_t|A_t, O^{(1)})$ will evaluate states to zero which satisfy $i = j\ \forall k$, this is because the measurement of object $O^{(1)}$ is independent of object $O^{(2)}$. If we had 3 objects, the joint would be $P(A_t = i, O^{(1)} = j, O^{(2)} = k, O^{(l)} = l)$ then the likelihood $P(Y_t|A_t, O^{(1)})$ will evaluate to zero for $i = j\ \forall k, l$ which would mean N^3 dependent states. In general, for M random variables the computational cost is $(M-1)N^{M-1}$ which gives $\mathcal{O}(N^{M-1})$ as opposed to the Histogram-SLAM's $\mathcal{O}(N^M)$. The computation complexity in this setup is still exponential but to the order $M-1$ as opposed to M which will nevertheless quickly limits the scalability as more objects are added.

Computing the value of a dependent state (i, j) in the joint distribution required evaluating Equation 5.4.4 which contains a product of t likelihood functions. However the likelihood functions are not overlapping and binary. As a result complete product does not have to be evaluated since only one likelihood function will effect the state (i, j). Thus evaluating Equation 5.4.4 yields a cost of $\mathcal{O}(1)$ and **not** $\mathcal{O}(t)$.

5.4.5 Scalable extension to multiple objects

To make the MLMF filter scalable we introduce an **independence assumption** between the objects and model the joint distribution (Equation 5.4.15) as a product of agent-object joint distributions:

$$P(A_t, O^{(1)}, \cdots, O^{(M-1)}|Y_{0:t}, u_{1:t}) = \prod_{i=1}^{M-1} P(A_t^{(i)}, O^{(i)}|Y_{0:t}^{(i)}, u_{1:t}) \tag{5.4.15}$$

The measurement variable Y_t, is the vector of all agent-object measurements, $Y_t = \left[Y_t^{(1)}, \ldots, Y_t^{(M-1)}\right]^{\mathrm{T}}$. Each agent-object joint distribution has its own parametrisation of the agent's marginal, $A_t^{(1)}, \ldots, A_t^{(M-1)}$ which combine to give the overall marginal of the agent A_t.

The computation of each object marginal $P(O^{(i)}|Y_{0:t}^{(i)})$ is independent of the other objects. This is evident from the marginalisation see Equation 5.4.16-5.4.18.

$$P(O^{(i)}|Y_{0:t}^{(i)}) = \sum_{A_t^{(i)}} P(A_t^{(i)}, O^{(i)}|Y_{0:t}^{(i)}) \tag{5.4.16}$$

$$P(A_t|Y_{0:t}, u_{1:t}) = \prod_{i=1}^{M-1} \left(\sum_{O^{(i)}} P(A_t^{(i)}, O^{(i)}|Y_{0:t}^{(i)}, u_{1:t}) \right) \tag{5.4.17}$$

$$= \prod_{i=1}^{M-1} P(A_t^{(i)}|Y_{0:t}^{(i)}, u_{1:t}) \tag{5.4.18}$$

The independence assumption will create an unwanted effect with respect to agent's marginal $P(A_t|Y_{0:t}, u_{1:t})$. At initialisation the agent marginals should be equal, $P(A_0|Y_0) = P(A_0^{(i)}|Y_0^{(i)})\forall_i$, however this is not the case because of Equation 5.4.18. To overcome this we define the final marginal, $P(A_t|Y_{0:t}, u_{1:t})$, of the agent as being the average of all the individual pairs $P(A^{(i)}|Y_{0:t}^{(i)}, u_{1:t})$.

$$P(A_t|Y_{0:t}, u_{1:t}) := \frac{1}{M-1} \sum_{i=1}^{M-1} P(A_t^{(i)}|Y_{0:t}^{(i)}, u_{1:t}) \tag{5.4.19}$$

Figure 5.13 (*Right*), depicts the graphical model of the scalable formulation. As each joint distribution pair has its own parametrisation of the agent's marginal and these do not subsequently get updated by one another, the information gained by one joint distribution pair is **not transferred**. A remedy is to transfer information between the marginals $A^{(i)}$ at specific intervals namely when one of the objects is sensed by the agent.

Algorithm 4: Scalable-MLMF: Measurement Update

```
input : $P(A_t^{(i)}|u_{1:t})$, $P(A_t^{(i)}|Y_{0:t-1}^{(i)}, u_{1:t})$
        $P(O^{(i)})$, $P(O^{(i)}|Y_{0:t-1}^{(i)}, u_{1:t})$
        $Y_t^{(i)}$
        $i = 1, \cdots, M$

▷ If object $i$ has been sensed by the agent
if $Y_t^{(i)} == 1$ then
    $P(O^{(i)}|Y_{0:t}^{(i)}) \leftarrow P(O^{(i)}|Y_{0:t-1}^{(i)})$ ;      ▷ measurement update Algo. 3
    $P(A_t^{(i)}|Y_{0:t}^{(i)}, u_{1:t}) \leftarrow P(A_t^{(i)}|Y_{0:t-1}^{(i)}, u_{1:t})$
    forall the $j \in (1, \dots M-1) \setminus i$ do
        $P(A_t^{(j)}|Y_{0:t}, u_{1:t}) = P(A_t^{(i)}|Y_{0:t}, u_{1:t})$
        $P(A_t^{(j)}|u_{1:t}) = P(A_t^{(i)}|u_{1:t})$
        $P(O^{(j)}|Y_{0:t}^{(i)}) \leftarrow \sum_{A^{(j)}} P(A_t^{(j)}, O^{(j)}|Y_{0:t}^{(i)})$ ;      ▷ object $j$ marginal
else
    forall the $i \in (1, \dots M)$ do
        measurement update Algo. 3
```

The exchange of information of one joint distribution to another is achieved through the agent's marginals $A^{(i)}$ according to Algorithm 4. The measurement update is the same as previously described in Algorithm 3 in the case of no positive measurements of the objects. If the agent senses an object, all of the agent marginals of the remaining joint distributions are set to the marginal of the joint distribution pair belonging to the positive measurement $Y_t^{(i)}$.

Figure 5.14, depicts the process of information exchange between object $O^{(1)}$ and $O^{(2)}$ in the event that the agent gets a positive sensation of $O^{(2)}$. Prior to the positive detection both marginals $P(A_t^{(1)}|Y_{0:t-1}^{(1)}, u_{1:t})$ and $P(A_t^{(2)}|Y_{0:t-1}^{(2)}, u_{1:t})$ occupy the same region and are identical. When the agent senses $O^{(2)}$ the line defined by the measurement likelihood function $P(Y_t^{(2)}|A_t^{(2)}, O^{(2)})$ becomes

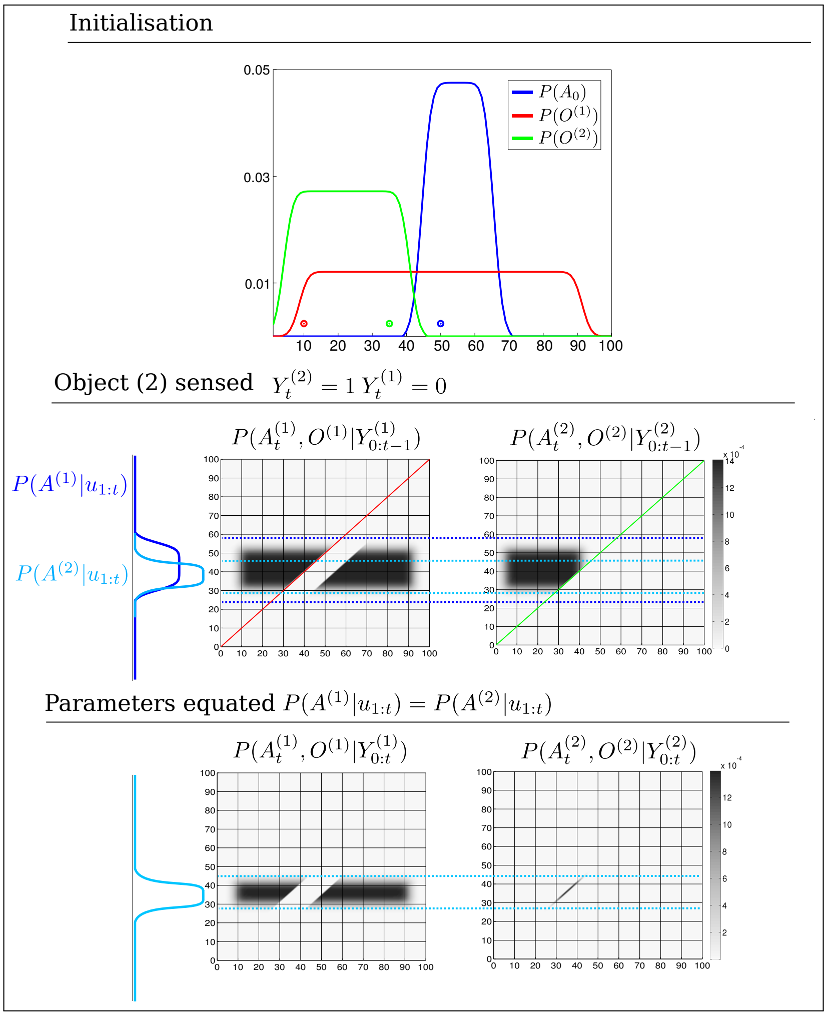

Figure 5.14: Transfer of information between joint distributions *Top:* Initial beliefs of the agent and object's location. The agent moves to the left until it senses object $O^{(2)}$. *Middle Left and Right:* Joint distributions of $P(A_t^{(1)}, O^{(1)} | Y^{(1)})$ and $P(A_t^{(2)}, O^{(2)} | Y^{(2)})$ prior sensing, $Y_t^{(2)} = 1$. The red and green lines across the joint distributions correspond to the region in which the measurement functions $P(Y_t^{(1)} | A_t^{(1)}, O^{(1)})$ and $P(Y_t^{(2)} | A_t^{(2)}, O^{(2)})$ will change the joint distributions. The dotted blue lines are for ease of comparing the joint distributions prior and post sensing. *Bottom right:* After the agent has sensed $O^{(2)}$, all the probability mass which was not overlapping the green line becomes an infeasible solution to the agent and object locations. At this point the marginals $P(A_t^{(1)} | u_{1:t}) \neq P(A_t^{(2)} | u_{1:t})$ are no longer equal (see the blue marginals *Middle left*). *Bottom left:* The constraint imposed by the measurement likelihood function of the second object (green line) is transferred to the joint distribution of the first object according to Algorithm 4. The result is a change in the joint distribution $P(A_t^{(1)}, O^{(1)} | Y^{(1)})$, which satisfies the constraints imposed by the agent's marginal from the joint distribution $P(A_t^{(2)}, O^{(2)} | Y^{(2)})$.

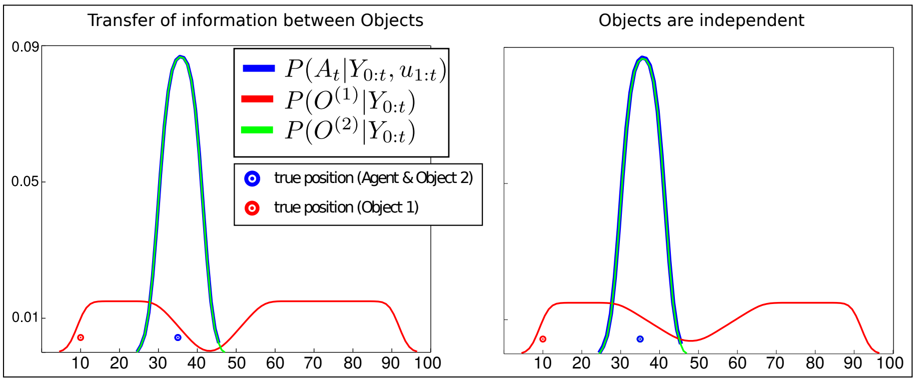

Figure 5.15: Independence & Objects *Left:* resulting marginals after setting the agent marginals of each joint distribution equal $A_t^{(1)} = A_t^{(2)}$ according to Algorithm 4. The object marginal $P(O^{(2)}|Y_{0:t})$ is recomputed. *Right:* resulting marginals in which the objects have no influence on one another. Note that a transfer of information has caused a change in the marginal $O^{(1)}$.

	space	**time**
Histogram	$\mathcal{O}(N^M)$	$\mathcal{O}(N^M)$
MLMF	$\mathcal{O}(M\,N)$	$\mathcal{O}(N^{(M-1)})$
scalable-MLMF	$\mathcal{O}(M\,N)$	$\mathcal{O}(M\,N)$

Table 5.2: Time and space complexity summary For both MLMF and scalabe-MLMF the worst case scenario is reported for the space complexity.

a hard constraint implying that both the agent and $O^{(2)}$ have to satisfy this constraint.

Figure 5.15 shows marginals resulting from the joint distributions in Figure 5.14. The marginals in the *Left* plot are the result after updating the marginals $A^{(i)}$. The *Right* plot shows the result for the case where the objects remain independent.

The result of introducing a dependency between the objects through the agent's marginals in the event of a sensing and treating them independently gives the same solution as the histogram filter in this particular case. However as each individual marginal $A_t^{(i)}$ diverges from the other marginals the filtered solution will diverge from the histogram's solution. We assume however that the objects are weakly dependent and sharing information during positive sensing events is sufficient. In section 5.5.2 we will evaluate the independence assumption with respect to the histogram filter.

Table 5.2 summarises the time and space complexity for the three filters.

5.5 Evaluation

We conduct three different types of evaluation to quantify the scalability and correctness of the scalable-MLMF filter. The first experiment tests the scalability of our filter in terms of processing time taken per motion-measurement

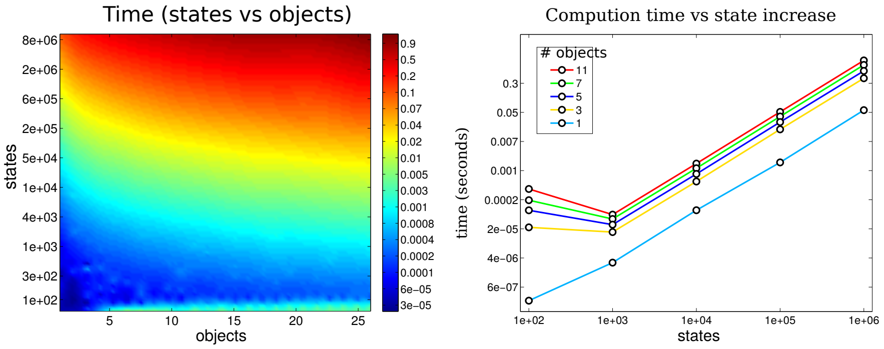

Figure 5.16: Time complexity: *left:* mean time taken for a loop update (motion and measurement) as a function of the number of states in a marginal and the number of objects present. *right:* time taken for a loop update with respect to the number of states in the marginal. The colour coded lines are associated with the number of objects present. The computational cost is plotted on a log scale. As the number of states increases exponentially the computational cost matches it.

update cycle. The second experiment evaluates the independence assumption between the objects which was made in the scalable-MLMF filter. The third and final experiment determines the effect the memory size has on a search policy to locate all the objects in the *Table* world.

5.5.1 Evaluation of time complexity

We measured the time taken by the motion-measurement update loop, as a function of the number beliefs and number of states per belief. We started with a 100 states per belief and gradually increase it to 10'000'000 over 50 steps. Each of the 50 steps treated 2 to 25 objects. Figure 5.16 *left* illustrates the computational cost as a function of number of states and objects. For each state-object pair 100 motion-measurement updates were performed. Most of the trials returned time updates below 1 Hz. Figure 5.16 *right* shows the computational cost as a function of the number of states plotted for 6 different filter runs with a different number of objects. As the number of states increases exponentially so does the computational cost. Note the cost increases at the same rate as the number of states meaning that the computational complexity is linear with respect to the number of states. This result is in agreement with the asymptotic time complexity.

5.5.2 Evaluation of the independence assumption

In section 5.4.5 we made the assumption (for scalability reasons) that the objects' beliefs are independent of one another. This assumption is validated by comparing the MLMF filter on three random variables, an agent and two objects, with the ground truth which we obtain from the standard histogram filter. For each of the three beliefs (the agent and two objects), 100 different marginals were generated and the true locations (actual position of the agent and objects) were sampled. Figure 5.17 *Top-left* illustrates one instance of the initialisation of the agent and object marginals with their associated sampled true position. The agent carried out a sweep of the state space for each of the marginals and the policy was saved and run with the scalable-MLMF filter. In the first experiment we assumed that the objects are completely independent and that there was no transfer of information between the pair-wise joint distributions. In the second and third experiments there is an exchange of information as described in Algorithm 4. Here we compare the effect of using the product of the agent's marginals, Equation 5.4.18, with the average of the marginals, Equation 5.4.19. We expect the average of the the agent's marginal to yield a closer result to the ground truth as the marginal of the agent $P(A_t|Y_{0:t}, u_{1:t})$ at initialisation will be the same as the ground truth (the Histogram-SLAM's). As for the marginal of the objects $P(O^{(i)}|Y_{0:t})$ we expect the difference between them to be independent of whether we use the product or average of the agents marginal. This is a result from Algorithm 4, when an object i is sensed all the corresponding agent marginals $P(A^{(j)}|u_{1:t})$ are set equal to $P(A^{(i)}|u_{1:t})$ and not to $P(A_t|Y_{0:t}, u_{1:t})$. This was a design decision of our information transfer heuristic, there are many other possibilities but this is one of the simplest. For each of the 100 sweeps the ground truth is compared with the scalabe-MLMF using the Hellinger distance (Equation 5.5.1)

$$H(P,Q) = \frac{1}{\sqrt{2}} \|\sqrt{P} - \sqrt{Q}\|_2 \tag{5.5.1}$$

which is a metric which measuring the distance between two probability distributions. Its value lies strictly between 0 (the two distributions are identical) and 1 (no overlap between them). Figure 5.17 shows the kernel density distribution of the Hellinger distances taken at each time step for all 100 sweeps. In the *Top-left* of the figure, for the case when no transfer of information is a applied, we can see that all the marginals are fare from the ground truth. This is the result of the introduction of the independence assumption, which was necessary for the MLMF to scale. In the *Bottom* of the figure we see the results for difference between the product and average of the agents marginals. As expected there is no difference between the objects marginals when considering both methods (product and average) with respect to the ground truth. The predominant difference is with respect to the agent's marginal $P(A_t|Y_{0:t}, u_{1:t})$. This was also expected and was reason behind the introduction of the average method instead of the product.

The scalable-MLMF information exchange heuristic will not lead to any probability mass of the objects marginals being falsely removed during the information transfer. The approach of the heuristic is close in spirit to a winner-

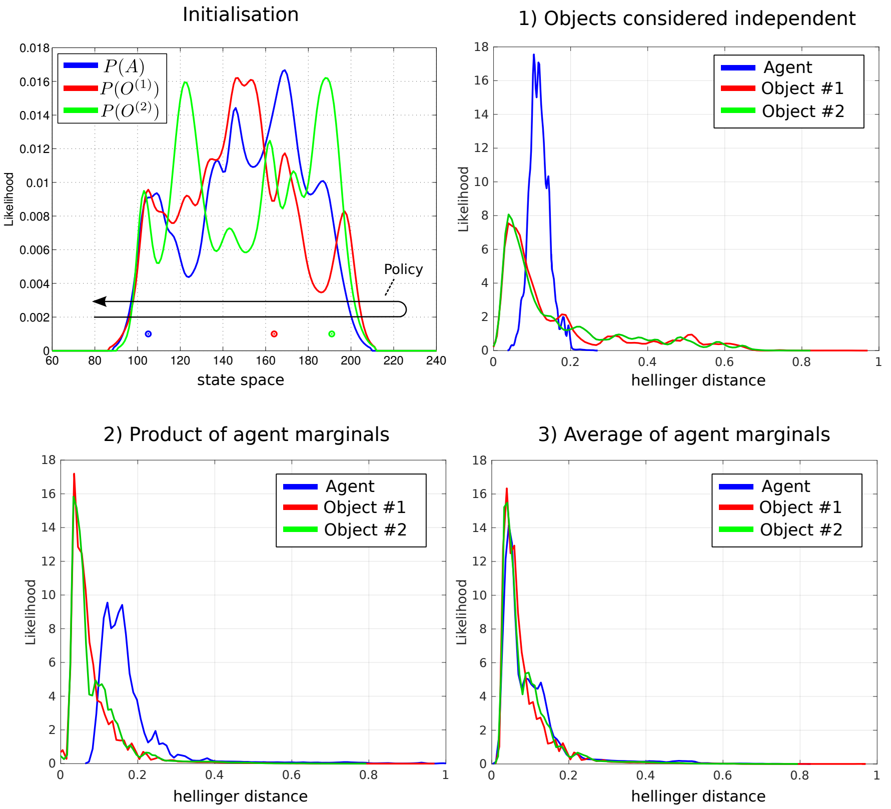

Figure 5.17: Comparison of scalable-MLMF and the histogram filter A deterministic sweep policy was carried out for 100 different initialisations of the agent and object beliefs. *Top left:* One particular Initialisation of the agent and object random variables. The true position of the agent and objects were sampled at random. The black arrow indicates the general policy which was followed for each of the 100 sweeps. These were performed for **1)** scalable-MLMF with objects considered to be independent at all times (no Algorithm 4). **2)** Agent marginal $P(A_t|Y_{0:t}, u_{1:t})$ is the product of marginals $P(A_t^{(i)}|Y_{0:t}^{(i)}, u_{1:t})$, Equation 5.4.18. **3)** marginal $P(A_t|Y_{0:t}, u_{1:t})$ is taken to be the average of all marginals $P(A_t^{(i)}|Y_{0:t}^{(i)}, u_{1:t})$, Equation 5.4.19. For each of these three experiment we report the kernel density estimation over the Hellinger distances taken at every time step between ground truth (from histogram filter) and scalable-MLMF.

take-all approach in terms of beliefs. When object i is sensed the agent marginal associated to it gets set to all other agent-object joint pairs, which results in the information accumulated in the jth agent marginals being replaced by the ith.

5.5.3 Evaluation of memory

The memory measurement likelihood function $P(Y_{0:t}|A_t, O, u_{1:t}; \Psi_{0:t})$ is parameterised by the history of all the measurement likelihood functions which have been applied on the joint distribution since initialisation. As detailed previously there can be no more than $|\Psi_{0:t}| \leq N$ different measurement likelihood functions added to memory. In the case of a very large state space this might

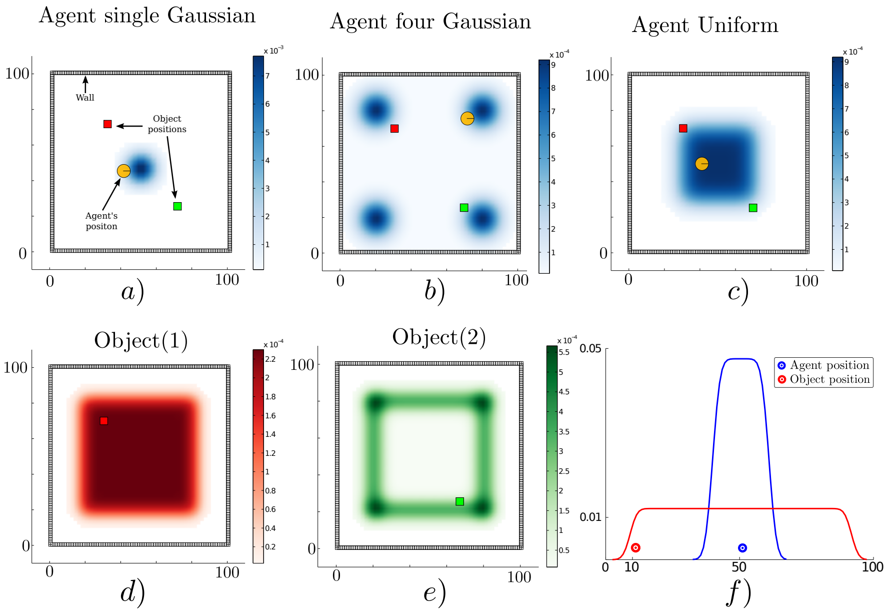

Figure 5.18: Agent's prior beliefs. Two types of environment, the first is a 2D world where the agent lives in a square surrounded by a wall whilst the second is a 1D world. In the 2D figures the agent is illustrated by a circle with a bar to indicate its heading. The true location of the objects are represented by colour coded squares. *Top row* three different initialisations of the agent's location. *Bottom row* d) the agent's prior beliefs with respect to the location of the first object and e) belief of the second object's location. *bottom row* f) 1D world with one object.

be cumbersome. We investigate how restricting the memory size, the number of parameters $|\Psi_{0:t}|$, can impact on the decision process in an Active-SLAM setting. Given our set up we choose a breadth-first search in the action space with a one time step horizon, making it a greedy algorithm. The objective function we utilise is the information gain of the beliefs after applying an action, Equation 5.5.2.

$$u_t = \arg\max_{u_t} H\{P(A_{t-1}, O|Y_{0:t-1}, u_{1:t-1})\} - \mathbb{E}_{Y_t}\left[H\{P(A_t, O|Y_{0:t}, u_{1:t})\}\right] \tag{5.5.2}$$

For each action we run the filter forward in time and consider all future measurements since we cannot know ahead of time the actual measurement. The information gain is the difference between the current entropy (defined by $H\{\cdot\}$) and the future entropy after the simulated motion and measurement update. The action with the highest information gain is subsequently selected. This is repeated at each time step. Figure 5.18 illustrates the environment setup for a 1D and 2D case. The agent's task is to find the objects in the environment.

For the 2D search we consider three different initialisations (single-Gaussian, four-Gaussian, Uniform) for the agent's belief where there are two objects to be found. Ten searches were carried out for each of the three initialisations of the agent's beliefs. The agent's true location, for each search, is sampled from its initial belief, and the objects' locations (red and green squares in Figure 5.18) are kept fixed throughout all searches. We repeated each search for 18 different memory sizes ranging from 1 to N (the number of states). For the 1D search case we considered one object since adding more objects would make the search easier. We are interested in how the memory effects the search and not the search itself. In Figures 5.19-5.20 we report on the time taken to find all objects with respect to a given memory size which is shown as the percentage of the total number of states. In the 1D search case the variability of time taken to find the object converges when the memory size is at 60% of the original state space. As for the 2D search with 2 beliefs (agent & 1 object) the convergence depends on the agent's initial belief. For the 1-Gaussian (green line) all searches take approximately the same amount of time after a memory size of 9%. As for the remaining two initialisations convergence is achieved at 48%. The same holds true for the case of 3 beliefs (agent & 2 objects).

In the 2D searches, the memory size has a less impact on the time taken to find the objects than in the 1D (which is a special search case). Only when the memory size is less than 6% is there a significant change. We conclude that at least in the case of the greedy one step-look ahead planner which is frequently used in the literature, the size of the memory seems not to be a limiting factor in terms of the time taken to accomplish the search.

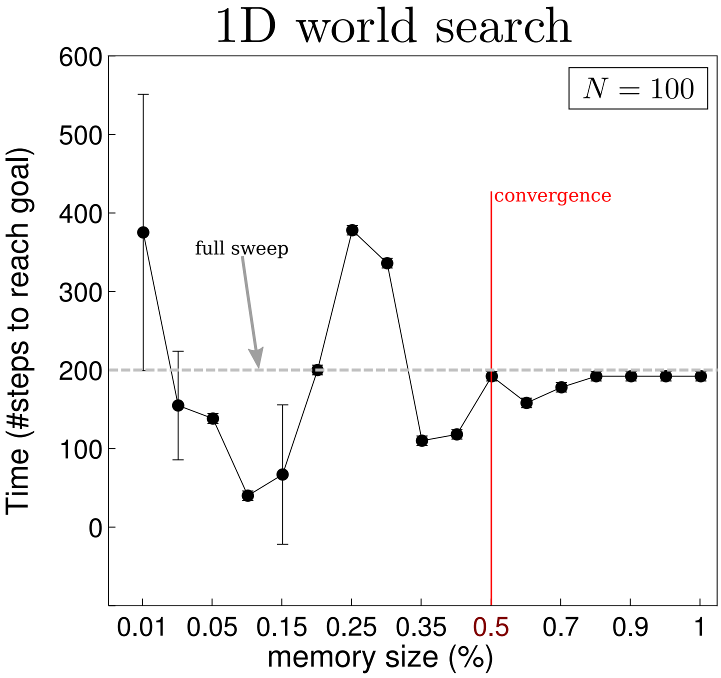

Figure 5.19: Memory size vs time to find object in 1D Results of the effect of the memory size on the decision process for the 1D search illustrated in Figure 5.18 *f)*. The memory size is reported as the percentage of total number of states present in the marginal space. At 100% the size of the memory is equal to that of the state space, $N = 100$ in this case. A total sweep of the entire state space would result in a total of 200 steps, the dotted grey line in the above figure. When no restrictions are placed on the memory size the policy following the greedy approach takes around 180 steps. This is result converges when number of parameters of the memory likelihood function is allowed to be bigger than 50% of the original state space.

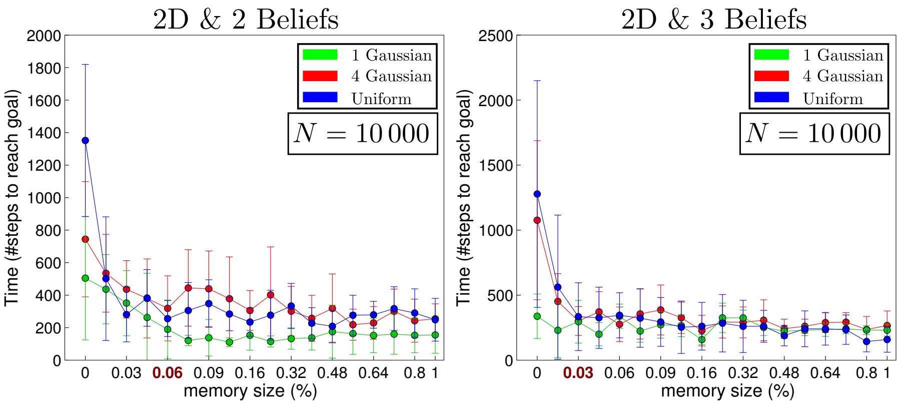

Figure 5.20: Memory size vs time to find objects in 2D. The initial beliefs correspond to does in Figure 5.18, a) for Gaussian (green line), b) 4 Gaussians (red line) and c) Uniform (blue line), both objects are initialised according to d) and e).

5.6 Conclusion

This work addresses the Active-SLAM filtering problem for scenarios in which sensory information relating to the map is very limited. Current SLAM algorithms filter the errors originating from sensory measurements and not prior uncertainty. By making the assumption that the joint distribution of all the random variables is a multivariate Gaussian, inference is tractable. Since the origin of our uncertainty does not originate from the measurement noise, we cannot assume the joint distribution to have any special structure. A suitable filter in this case would be the histogram which makes no assumption about the shape or form taken by the joint distribution. However, the space and time complexity are exponential with respect to the number random variables and this is a major limiting factor for scalability.

The main contribution of this work is a formulation of a histogram Bayesian state space estimator in which the computational complexity is both linear in time and space. We have taken a different approach to other SLAM formulations in the sense that we did not explicitly parameterise the joint distribution but kept all parameterisation in the space of the marginals (beliefs), avoiding the exponential increase in parameter space which would otherwise have been the case. Our filter's parameters consist of the marginals and the history of measurement functions which have been applied. By solely evaluating the joint distribution at the states which are affected by the current measurement function whilst taking into account the memory, the MLMF filter obtains the same filtered marginals as the histogram filter. Further, the worst case space complexity is linear rather than exponential and the time complexity remains exponential but increases at lower rate than in the histogram filter. In striving to make the filter scalable we assume an independence assumption between the objects. An individual MLMF filter was used for each agent-object pair. We evaluated the difference between the scalable-MLMF filter with a ground truth provided by the histogram filter for 100 different searches with respect to the Hellinger distance. We conclude that the divergence was relatively small and thus the scalable-MLMF filter provides a good approximation to the true filtered marginals. We evaluated the time taken to perform a motion-update loop for different discretisations of the state space (100 to 10'000'000 states) and number of objects (2 to 25). In most of the cases we achieved an update cycle rate below 1Hz. We evaluated how the increase of the number of states effected the computational cost. The relationship was found to be linear and thus in agreement with our analysis of the asymptotic growth rate. We analysed the effect of the memory size (the remembered number of measurement likelihood functions) on the decision theoretic process of reducing the uncertainty of the map and agent during a search task. We found that in the 2D case the memory size has much less effect than in the 1D case and we conclude that it is unnecessary to remember every single measurement function.

The implications of the MLMF and scalable-MLMF filter are that we have a computationally tractable means of performing SLAM in a case scenario in which mostly negative information is present and the joint distribution cannot be assumed to have any specific structure. The filter can be used at a higher cognitive level than processing raw sensory information as it is often in Active-SLAM. It would be well suited for reasoning tasks where the robot's field of view is limited.

An interesting future extension would be to make the original MLMF filter scalable without introducing assumptions. One possibility could to be to consider Monte Carlo integration methods for inference. These can scale well to high dimensional spaces whilst still providing reliable estimates. A second possibility could be to investigate the use of Gaussian Mixtures as a form of parameterisation of the marginals so that it might be possible to blend our filter with EKF-SLAM. This would allow the parameters to grow quadratically with respect to the dimension of the marginal space as opposed to exponentially as it is in the case with the histogram and MLMF filters.

5.7 Appendix

5.7.1 Probabilities

There are two rules are extensively used in the derivation of a Bayesian Filter filter recursion regardless of the chosen parameterisation. Although well known we restate them here for convenience.

- **Chain rule**

$$P(X_M, \cdots, X_1) = P(X_M|X_{M-1}, \cdots, X_1)P(X_{M-1}, \cdots, X_1) \tag{5.7.1}$$

- **Conditional independence**

$$\begin{aligned} P(A, B|C) &= P(A|\cancel{B}, C)P(B|C) \\ &= P(A|C)P(B|C) \end{aligned} \tag{5.7.2}$$

A and B are independent given that C is known.

5.7.2 Bayesian filtering recursion

Joint distribution

The joint distribution is over all the random variables since $t = 0$ until the current time step t:

$$
\begin{aligned}
& P(A_{0:t}, O, Y_{0:t}|u_{1:t}) = \\
& P(A_0)P(O)P(Y_0|A_0, O)\prod_{t=1}^{t} P(A_t|A_{t-1}, u_t)P(Y_t|A_t, O) && (5.7.3)\\
& P(A_0)P(O)P(Y_0|A_0, O)P(A_{1:t}|u_{1:t})P(Y_{1:t}|A_{1:t}, O) && (5.7.4)\\
& P(A_0)P(O)P(A_{1:t}|u_{1:t})P(Y_{0:t}|A_{0:t}, O) && (5.7.5)
\end{aligned}
$$

From to we t[illegible]ansition and likelihood functions separately and made use of the chain rule of probabilities (Equation 5.7.1) and conditional independence

From Equation 5.7.5 by multiplying the first three terms together we get [illegible] $P(A_{0:t}$ [illegible] $_{1:t})$ and we can see clearly the relationship with the joint distribution:

$$P(A_{0:t}, O, Y_{0:t}|u_{1:t}) = P(A_{0:t}, O|u_{1:t})P(Y_{0:t}|A_{0:t}, O) \qquad (5.7.6)$$

Typically we are interested in the filtered joint distribution $P(A_t, O, Y_{0:t}|u_{1:t})$ which is obtained by marginalising over $A_{0:t-1}$:

$$P(A_t, O, Y_{0:t}|u_{1:t}) = \sum_{A_{0:t-1}} P(A_t, A_{0:t-1}, O, Y_{0:t}|u_{1:t}) \qquad (5.7.7)$$

A recursion exists making it not necessary to sum over all agents random variables from the state time until th[illegible]d, but instead only have to consider the last time step, see the next paragraph.

Filtering problem

We derive $P(A_t, O, Y_{0:t}|u_{1:t})$, we start from the joint distribution, Equation 5.7.8:

$$P(A_t, O, Y_{0:t}|u_{1:t}) = \sum_{A_{t-1}} P(A_t, A_{t-1}, O, Y_t, Y_{0:t-1}|u_t, u_{1:t-1}) \qquad (5.7.8)$$

$$
\begin{aligned}
P(A_t, O, Y_{0:t}|u_{1:t}) \quad = \textstyle\sum_{A_{t-1}} \quad & P(Y_t|\cancel{Y_{0:t-1}}, A_t, \cancel{A_{t-1}}, O, \cancel{u_t}, \cancel{u_{1:t-1}})\\
& P(A_t|A_{t-1}, \cancel{O}, u_t, \cancel{Y_{0:t-1}}, \cancel{u_{1:t-1}})\\
& P(A_{t-1}, O, Y_{0:t-1}|\cancel{u_t}, u_{1:t-1})
\end{aligned}
$$

$$
\begin{aligned}
& P(A_t, O, Y_{0:t}|u_{1:t}) && (5.7.9)\\
& = \sum_{A_{t-1}} P(Y_t|A_t, O)P(A_t|A_{t-1}, u_t)P(A_{t-1}, O, Y_{0:t-1}|u_{1:t-1})\\
& = P(Y_t|A_t, O)\underbrace{\sum_{A_{t-1}} P(A_t|A_{t-1}, u_t)P(A_{t-1}, O, Y_{0:t-1}|u_{1:t-1})}_{P(A_t, O, Y_{0:t-1}|u_{1:t})} && (5.7.10)
\end{aligned}
$$

$$\begin{aligned} P(A_t, O, Y_{0:t}|u_{1:t}) &= P(Y_t|A_t, O)P(A_t, O, Y_{0:t-1}|u_{1:t}) \\ &= P(Y_t|A_t, O)P(A_t, O|Y_{0:t-1}, u_{1:t})P(Y_{0:t-1}|u_{1:t}) \qquad (5.7.11) \end{aligned}$$

All the cancellations come from the *Markov Assumption* read from the structure of the Bayesian network. The resulting final Bayesian recursion is obtained by conditioning on the measurement and actions, which is the normalisation factor.

$$\begin{aligned} P(A_t, O|Y_{0:t}, u_{1:t}) &= \frac{P(Y_t|A_t, O)P(A_t, O|Y_{0:t-1}, u_{1:t})P(Y_{0:t-1}|u_{1:t})}{P(Y_{0:t}|u_{1:t})} \\ &= \frac{P(Y_t|A_t, O)P(A_t, O|Y_{0:t-1}, u_{1:t})\cancel{P(Y_{0:t-1}, u_{1:t})}}{P(Y_t|Y_{0:t-1}, u_{1:t})\cancel{P(Y_{0:t-1}|u_{1:t})}} \qquad (5.7.12) \end{aligned}$$

$$P(A_t, O|Y_{0:t}, u_{1:t}) = \frac{P(Y_t|A_t, O)P(A_t, O|Y_{0:t-1}, u_{1:t})}{P(Y_t|Y_{0:t-1}, u_{1:t})} \qquad (5.7.13)$$

The evidence is the the integration of all terms which are not measurements in the numerator of Equation 5.7.13.

$$\begin{aligned} P(Y_t|Y_{0:t-1}, u_{1:t}) &= \sum_{A_t}\sum_{O} P(Y_t|A_t, O)P(A_t, O|Y_{0:t-1}, u_{1:t}) \qquad (5.7.14) \\ &= \sum_{A_t}\sum_{O} P(Y_t, A_t, O|Y_{0:t-1}, u_{1:t}) \qquad (5.7.15) \end{aligned}$$

This is very expensive since we have to sum over the entire joint distribution, the MLMF avoids doing this by only considering the dependent regions of the joint distribution.

5.7.3 Recursion example

Derivation of the filtered joint distribution, $P(A_t, O, Y_t|Y_{0:t}, u_{1:t})$, for two updates. At initialisation when no action has yet been taken the filtered joint distribution is the product of the initial marginals and first likelihood function:

$$P(A_0, O, Y_0) = P(O)P(A_0)P(Y_0|A_0, O) \qquad (5.7.16)$$

The a first action, u_1 is applied, which to get the filtered joint distribution is marginalised:

$$\begin{aligned}
P(A_1, O, Y_0|u_1) &= P(O)\sum_{A_0} P(A_1|A_0, u_1)P(A_0)P(Y_0|A_0, O) && (5.7.17)\\
&= P(O)\sum_{A_0} P(A_1, A_0, Y_0|u_1, O) && (5.7.18)\\
&= P(O)P(A_1, Y_0|u_1, O) && (5.7.19)\\
&= P(O)P(Y_0|A_1, O, u_1)P(A_1|u_1, \cancel{O}) && (5.7.20)\\
&= P(O)P(Y_0|A_1, O, u_1)P(A_1|u_1) && (5.7.21)
\end{aligned}$$

From Equation 5.7.19 to 5.7.20 we used the Chain rule 5.7.1 and the cancellation in Equation 5.7.20 arise from the factorisation of the joint distribution, see Figure 5.3 on page 123, A's marginal does not depend on O. After the application of the first action, the filtered joint has the following form:

$$P(A_1, O, Y_0|u_1) = P(O)P(A_1|u_1)P(Y_0|A_1, O, u_1) \qquad (5.7.22)$$

A second measurement Y_1 and action u_2 are integrated into the filtered joint distribution:

$$\begin{aligned}
&P(A_2, O, Y_{0:1}|u_{1:2}) = \\
&P(O)\sum_{A_1} P(A_2|A_1, u_2)P(A_1|u_1)P(Y_0|A_1, O, u_1)P(Y_1|A_1, O) = \\
&P(O)\sum_{A_1} P(A_2, A_1|u_{1:2})P(Y_{0:1}|A_1, O, u_1) = \\
&P(O)\sum_{A_1} P(A_2, A_1, Y_{0:1}|O, u_{1:2}) = \\
&P(O)P(A_2, Y_{0:1}|O, u_{1:2}) = && (5.7.23)\\
&P(O)P(Y_{0:1}|A_2, O, u_{1:2})P(A_2|\cancel{O}, u_{1:2}) && (5.7.24)
\end{aligned}$$

We expand the function $P(Y_{0:1}|A_2, O, u_{1:2})$ to give a sense of how the likelihood function's positions get as illustrated in Figure 5.8 on page 129.

$$\begin{aligned}
P(Y_0, Y_1|A_2, O, u_1, u_2) &= P(Y_0|\cancel{Y_1}, A_2, O, u_1, u_2)P(Y_1|A_2, O, \cancel{u_1}, u_2) && (5.7.25)\\
&= P(Y_0|A_2, O, u_{1:2})P(Y_1|A_2, O, u_2) && (5.7.26)
\end{aligned}$$

The first likelihood of measurement Y_0 is dependent on the last to applied actions whilst the likelihood of Y_1 is dependent on the last action.

Repeating the above for $Y_{2:t}$ and $u_{3:t}$ results in:

$$P(A_t, O, Y_{0:t}|u_{1:t}) = P(O)P(A_t|u_{1:t})\prod_{i=0}^{t} P(Y_i|A_t, O, u_{i:t}) \qquad (5.7.27)$$

where we defined $P(Y_0|A_0, O, u_0) := P(Y_0|A_0, O)$ and $P(A_0|u_{1:0}) := P(A_0)$.

5.7.4 Derivation of the evidence

The evidence, also known as the marginal likelihood, is the marginalisation of all non measurement random variables from the filtered joint distribution $P(A_t, O, Y_{0:t}|u_{1:t})$. We detail below how we compute the evidence in a recursive manner whilst only considering dependent regions of the joint distribution.

We start with the definition of the evidence (we dropped $u_{1:t}$ to simplify the notation).

$$P(Y_{0:t}) = \sum_{A_t} \sum_{O} P(A_t, O, Y_{0:t}) \tag{5.7.28}$$

As we are interested in a recursive computation of the evidence, we consider the gradient:

$$\alpha_t = \nabla_{Y_t} P(Y_{0:t}) = P(Y_{0:t}) - P(Y_{0:t-1}) \tag{5.7.29}$$

$$\begin{aligned}
\alpha_t &= \sum_{A_t} \sum_{O} P(A_t, O, Y_{0:t}) - P(A_t, O, Y_{0:t-1}) && (5.7.30)\\
&= \sum_{A_t} \sum_{O} P(Y_t|A_t, O) P(A_t, O, Y_{0:t-1}) - P(A_t, O, Y_{0:t-1}) && (5.7.31)\\
&= \sum_{A_t} \sum_{O} (P(Y_t|A_t, O) - 1) P(A_t, O, Y_{0:t-1}) && (5.7.32)
\end{aligned}$$

The gradient α_t is the difference in mass before and after the application the likelihood function, $P(Y_t|A_t, O)$. The above summation, Equation 5.7.32, is over the entire joint distribution state space. We can take advantage of the fact that the likelihood function is sparse and will only affect a small region of the joint distribution, which we called the dependent states, $\cap$. The states which are not affected by the joint distribution will result in a contribution of zero to Equation 5.7.32. We rewrite the gradient update in terms of only the dependent regions:

$$\alpha_t = \sum_{A_t} \sum_{O} (P(Y_t|A_t, O) - 1) P_{\cap}(A_t, O, Y_{0:t-1}) \tag{5.7.33}$$

Consider the first update of the evidence at time $t = 0$:

$$\alpha_0 = \sum_{A_t} \sum_{O} (P(Y_0|A_0, O) - 1) P(A_0, O) \tag{5.7.34}$$

The one in Equation 5.7.35 is the original value of the normalisation denominator before any observation is made and as the initial joint distribution $P(A_0, O)$ is normalised the value of the denominator is one.

$$P(Y_0) = 1 + \alpha_0 \tag{5.7.35}$$

For the evidence $P(Y_{0:t})$ we consider the summation of all the derivatives α_t from time $t = 0$ until t:

$$P(Y_{0:t}) = 1 + \sum_{t=0}^{t} \alpha_t \tag{5.7.36}$$

5.7.5 Derivation of the marginal

The marginal of a random variable is the marginalisation or integration over all other random variables, $P(A_t, |Y_{0:t}) = \sum_O P(A_t, O|Y_{0:t})$. Below we give a form of this integration which exploits the independent regions in the joint distribution.

$$P(A_t, |Y_{0:t}) = \mathbf{P(A_t|Y_{0:t-1})} - \Big(\mathbf{P(A_t|Y_{0:t-1})} - P(A_t|Y_{0:t})\Big) \tag{5.7.37}$$

In Equation 5.7.37 we add and substract $P(A_t|Y_{0:t-1})$ and we further split $P(A_t|Y_{0:t-1})$ into independent and dependent components:

$$\begin{aligned} &P(A_t, |Y_{0:t}) = P(A_t|Y_{0:t-1}) - \\ &\Big(\underbrace{P_\cap(A_t|Y_{0:t-1}) + \cancel{P_\ominus(A_t|Y_{0:t-1})}}_{P(A_t|Y_{0:t-1})} - \underbrace{P_\cap(A_t|Y_{0:t}) + \cancel{P_\ominus(A_t|Y_{0:t})}}_{P(A_t|Y_{0:t})}\Big) \end{aligned} \tag{5.7.38}$$

From equation 5.7.38 to 5.7.39 we used the fact that independent regions of the marginal distributions will remain unchanged after an observation, $P_\ominus(A_t|Y_{0:t-1}) = P_\ominus(A_t|Y_{0:t})$, and before re-normalisation. This results in the final recursive update:

$$P(A_t, |Y_{0:t}) = P(A_t|Y_{0:t-1}) - \Big(P_\cap(A_t|Y_{0:t-1}) - P_\cap(A_t|Y_{0:t})\Big) \tag{5.7.39}$$

Equation 5.7.39 states that only elements of the marginals which are dependent will change by the difference of before and after a measurement update.