University of Rome “La Sapienza”

Master in Artificial Intelligence and Robotics

Probabilistic Robotics

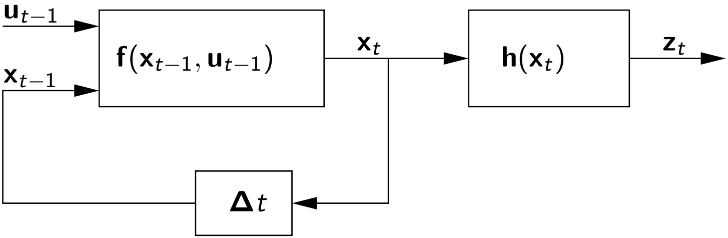
Prof. Giorgio Grisetti

Course web site:
http://www.dis.uniroma1.it/~grisetti/teaching/probabilistic_ro

Overview

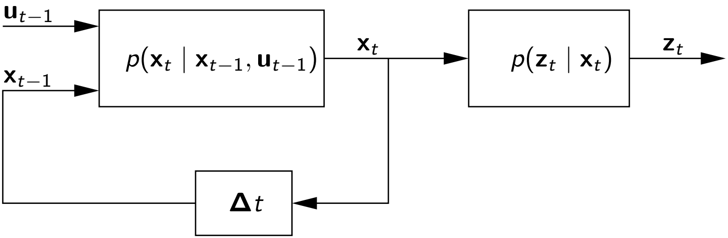
- Probabilistic Dynamic Systems
- Dynamic Bayesian Networks (DBN)
- Inference on DBN
- Recursive Bayes Equation

Dynamic System, Deterministic View



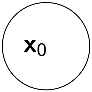
- $\mathbf{f}(\mathbf{x}_{t-1}, \mathbf{u}_{t-1})$: transition function
- $\mathbf{h}(\mathbf{x}_t)$: observation function
- $\mathbf{x}_{t-1}$: previous state
- $\mathbf{x}_t$: current state
- $\mathbf{u}_{t-1}$: previous control/action
- $\mathbf{z}_t$: current observation
- $\mathbf{\Delta}t$: delay

Dynamic System, Probabilistic View



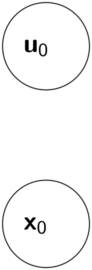
- $p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_{t-1})$: transition model
- $p(\mathbf{z}_t \mid \mathbf{x}_t)$: observation model
- $\mathbf{x}_{t-1}$: previous state
- $\mathbf{x}_t$: current state
- $\mathbf{u}_{t-1}$: previous control/action
- $\mathbf{z}_t$: current observation
- $\boldsymbol{\Delta} t$: delay

Evolution of a Dynamic System



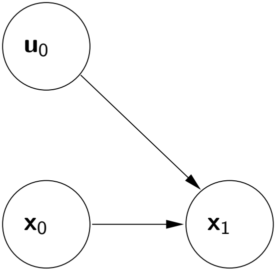
Let's start from a known initial state distribution $p(\mathbf{x}_0)$.

Evolution of a Dynamic System



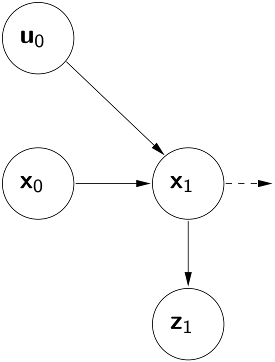
A control $\mathbf{u}_0$ becomes available

Evolution of a Dynamic System



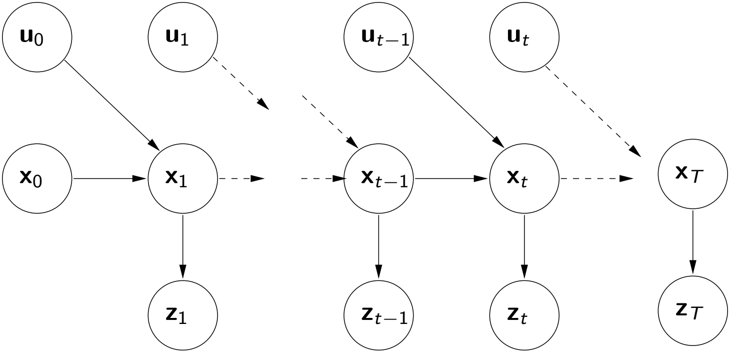
The transition model $p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_{t-1})$ correlates the current states $\mathbf{x}_1$ with the previous control $\mathbf{u}_0$ and the previous state $\mathbf{x}_0$.

Evolution of a Dynamic System



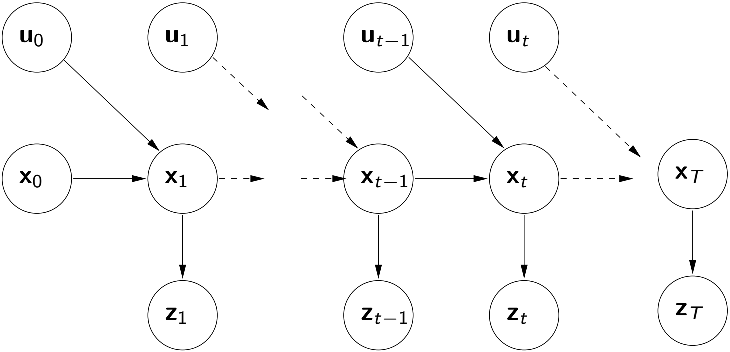
The observation model $p(\mathbf{z}_t \mid \mathbf{x}_t)$ correlates the observation $\mathbf{z}_1$ and the current state $\mathbf{x}_1$.

Evolution of a Dynamic System



This leads to a recurrent structure, that depends on the *time*.

Dynamic Bayesian Networks (DBN)



- Graphical representations of stochastic dynamic processes
- Characterized by a recurrent structure

States in a DBN

The domain of the states $\mathbf{x}_t$, the controls $\mathbf{u}_t$ and the observations $\mathbf{z}_t$ are not restricted to be boolean or discrete. Examples:

- Robot localization, with laser range finder
  - states $\mathbf{x}_t \in SE(2)$, isometries on a plane
  - observations $\mathbf{z}_t \in \Re^{\#\mathrm{beams}}$, laser range measurements
  - controls $\mathbf{u}_t \in \Re^2$, translational and rotational speed
- HMM
  - states $\mathbf{x}_t \in [X_1, \ldots, X_{N_x}]$, finite states
  - observations $\mathbf{z}_t \in [Z_1, \ldots, Z_{N_z}]$, finite observations
  - controls $\mathbf{u}_t \in [U_1, \ldots, U_{N_u}]$, finite observations

Inference in a DBN requires to design a data structure that can represent a *distribution* over states.

Typical Inferences in a DBN

In a dynamic system, usually[1] we know:

- the observations made by the system $\mathbf{z}_{1:T}$, because we *measure* them
- the controls $\mathbf{u}_{1:T-1}$ because we *issue* them

Typical inferences in a DBN:

name	query	known
Filtering	$p(\mathbf{x}_T \vert \mathbf{u}_{0:T-1}, \mathbf{z}_{1:T})$	$\mathbf{u}_{0:T-1}, \mathbf{z}_{1:t}$
Smoothing	$p(\mathbf{x}_k \vert \mathbf{u}_{0:T-1}, \mathbf{z}_{1:T}),\ 0 < k < T$	$\mathbf{u}_{0:T-1}, \mathbf{z}_{1:t}$
Max a Posteriori	$\mathrm{argmax}_{\mathbf{x}_{0:T}}\ p(\mathbf{x}_{0:T} \mid \mathbf{u}_{0:T-1}, \mathbf{z}_{1:T})$	$\mathbf{u}_{0:T-1}, \mathbf{z}_{1:T}$

[1] usually does not mean always

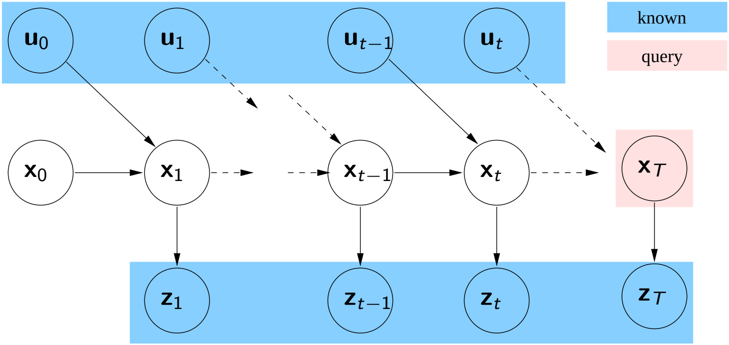
Typical Inferences in a DBN

Using the traditional tools for Bayes Networks is not a good idea.

- too many variables (potentially infinite) render the solution intractable
- the domains are not necessarily discrete

However, we can exploit the recurrent structure to design procedures that take advantage of it.

DBN Inference: Filtering

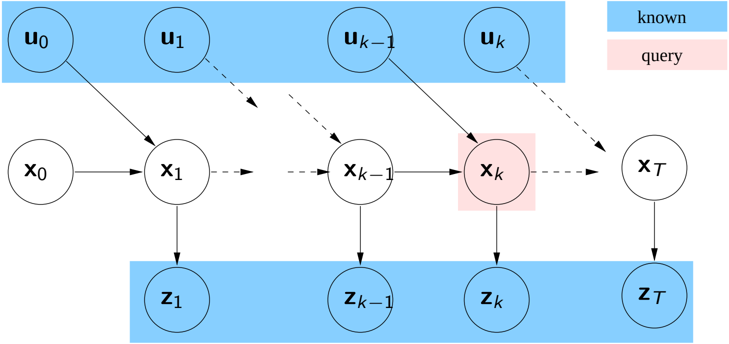


Given

- the sequence of all observations $\mathbf{z}_{1:T}$ up to the current time T,
- the sequence of all controls $\mathbf{u}_{0:T-1}$

we want to compute the distrubution over the current states $p(\mathbf{x}_T|\mathbf{u}_{0:T-1}, \mathbf{z}_{1:T})$.

DBN Inference: Smoothing

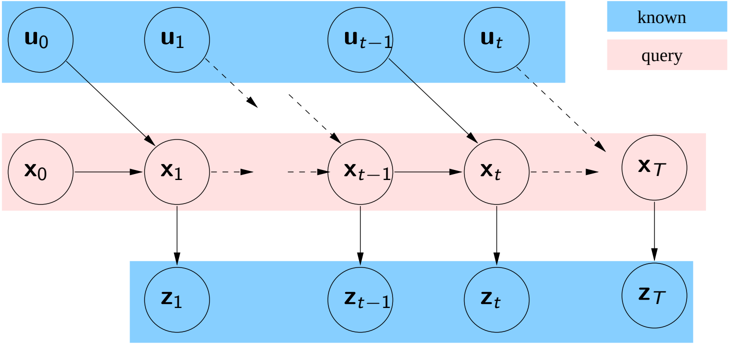


Given

- the sequence of all observations $\mathbf{z}_{1:T}$ up to the current time T,
- the sequence of all controls $\mathbf{u}_{0:T-1}$

we want to compute the distrubution over a past state $p(\mathbf{x}_k|\mathbf{u}_{0:T-1}, \mathbf{z}_{1:T})$. Knowing also the controls $\mathbf{u}_{k:T-1}$ and the observations $\mathbf{Z}_{k:T}$ *after* time k, leads to more accurate estimates than pure filtering.

DBN Inference: Maximum a Posteriori



Given

- the sequence of all observations $\mathbf{z}_{1:T}$ up to the current time T,
- the sequence of all controls $\mathbf{u}_{0:T-1}$

we want to find the most likely *trajectory* of states $\mathbf{x}_{0:T}$.
In this case we are not seeking for a distribution. Just the most likely sequence.

DBN inference: Belief

- Algorithms for performing inference on a DBN keep track of the *estimate* of a distribution of states.
- This distribution should be stored in an appropriate data structure.
- The structure depends on
  - the knowledge of the characteristics of the distribution (e.g. Gaussian)
  - the domain of the state variables (e.g. continuous vs discrete)

When we write $b(\mathbf{x}_t)$ we mean our current belief of $p(\mathbf{x}_t|\ldots)$.
The algorithms for performing inference on a DBN work by updating a belief.

DBN inference: Belief

- In the simple case of a system with discrete state $\mathbf{x} \in \{X_{1:n}\}$, the belief can be represented through and array $\mathbf{x}$ of float values. Each cell of the array $\mathbf{x}[i] = p(\mathbf{x} = X_i)$ contains the probability of that state.
- If our system has a continous state and we know it is distributed according to a Gaussian, we can represent the belief through its parameters (mean and covariance matrix).
- If the state is continuous but the distribution is unknown, we can use some approximate representation (e.g. weighed samples of state values).

Filtering: Bayes Recursion

We want to compute $p(\mathbf{x}_T \mid \mathbf{u}_{0:T-1}, \mathbf{z}_{1:T}) =?$
We know:

- the observations $\mathbf{z}_{1:T}$
- the controls $\mathbf{u}_{0:T-1}$
- $p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_{t-1})$: the transition model. It is a function that given the previous state $\mathbf{x}_{t-1}$ and control $\mathbf{u}_{t-1}$, tells us how likely it is to lend in state $\mathbf{x}_t$.
- $p(\mathbf{z}_t \mid \mathbf{x}_t)$: the transition model. It is a function, that given the current state $\mathbf{x}_{t-1}$, tells us how likely it is to observe $\mathbf{z}_t$.
- $b(\mathbf{x}_{t-1})$, which is our previous belief about $p(\mathbf{x}_{t-1}, \mid \mathbf{u}_{0:t-2}, \mathbf{z}_{1:t-1})$

Filtering (1)

$$p(\mathbf{x}_t \mid \mathbf{u}_{0:t-1}, \mathbf{z}_{1:t}) = \tag{1}$$

splitting $\mathbf{z}_t$:

$$= p(\underbrace{\mathbf{x}_t}_{A} \mid \underbrace{\mathbf{z}_t}_{B}, \underbrace{\mathbf{u}_{0:t-1}, \mathbf{z}_{1:t-1}}_{C}) \tag{2}$$

recall the conditional bayes rule $p(A|B,C) = \frac{p(B|A,C)p(A|C)}{p(B|C)}$

$$= \frac{p(\overbrace{\mathbf{z}_t}^{B} \mid \overbrace{\mathbf{x}_t}^{A}, \overbrace{\mathbf{u}_{0:t-1}, \mathbf{z}_{1:t-1}}^{C})\, p(\overbrace{\mathbf{x}_t}^{A} \mid \overbrace{\mathbf{u}_{0:t-1}, \mathbf{z}_{1:t-1}}^{C})}{p(\underbrace{\mathbf{z}_t}_{B} \mid \underbrace{\mathbf{u}_{0:t-1}, \mathbf{z}_{1:t-1}}_{C})} \tag{3}$$

Filtering: Denominator

- Let the denominator

$$\eta_t = 1/p(\mathbf{z}_t \mid \mathbf{u}_{0:t-1}, \mathbf{z}_{1:t-1}). \tag{4}$$

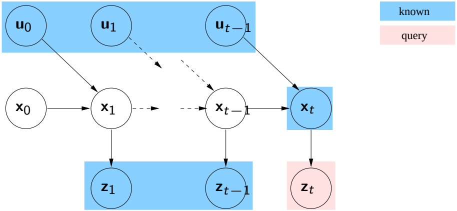
- Note that η_t does not depend on $\mathbf{x}$, thus to the extent of our computation is just a normalizing constant.
- We will come back to the denominator later.

Filtering: Observation model

- Our filtering equation becomes

$$\eta_t p(\mathbf{z}_t \mid \mathbf{x}_t, \mathbf{u}_{0:t-1}, \mathbf{z}_{1:t-1}) p(\mathbf{x}_t \mid \mathbf{u}_{0:t-1}, \mathbf{z}_{1:t-1}) \tag{5}$$

- Note that $p(\mathbf{z}_t \mid \mathbf{x}_t, \mathbf{u}_{0:t-1}, \mathbf{z}_{1:t-1})$ means this



If we know $\mathbf{x}_t$, we do not need to know $\mathbf{u}_{0:t-1}, \mathbf{z}_{1:t-1}$ to predict $\mathbf{z}_t$, since the state $\mathbf{x}_t$ encodes all the knowledge about the past (Markov assumption) :

$$p(\mathbf{z}_t \mid \mathbf{x}_t, \mathbf{u}_{0:t-1}, \mathbf{z}_{1:t-1}) = p(\mathbf{z}_t \mid \mathbf{x}_t) \tag{6}$$

Filtering: Transition Model

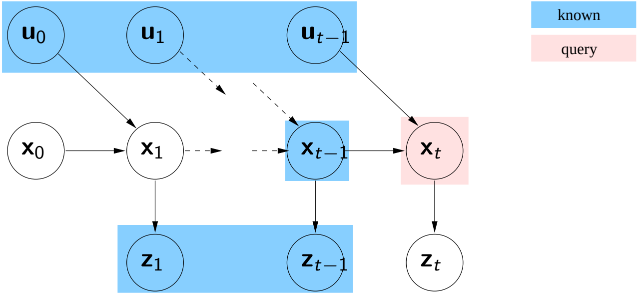
- Thus, our current equation is

$$p(\mathbf{x}_t \mid \mathbf{u}_{0:t-1}, \mathbf{z}_{1:t}) = \eta_t p(\mathbf{z}_t \mid \mathbf{x}_t) p(\mathbf{x}_t \mid \mathbf{u}_{0:t-1}, \mathbf{z}_{1:t-1}) \tag{7}$$

- Still the second part of the equation is obscure.
- Our task is to manipulate it to get something that matches our preconditions.

Filtering: Transition Model

- If we would know $\mathbf{x}_{t-1}$, our life would be nuch easier, as we could repeat the trick done for the observation model:



- thus

$$p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_{0:t-1}, \mathbf{z}_{1:t-1}) = p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_{t-1}) \quad (8)$$

Filtering: Transition Model

The sad truth is that we do not have $\mathbf{x}_{t-1}$, however

- Recall the following probability identities

$$p(A|C) = \sum_b p(A, B|C) \tag{9}$$

$$p(A, B|C) = p(A|B, C)p(B|C) \tag{10}$$

- combining the two above we have

$$p(A|C) = \sum_b p(A|B, C)p(B|C) \tag{11}$$

Filtering: Transition Model

The sad truth is that we do not have $\mathbf{x}_{t-1}$, however

- let's look again at our problematic equation, and put some letters

$$p(\underbrace{\mathbf{x}_t}_{A} \mid \underbrace{\mathbf{u}_{0:t-1}, \mathbf{z}_{1:t-1}}_{C}) = \tag{12}$$

$$= \sum_{\mathbf{x}_{t-1}} p(\underbrace{\mathbf{x}_t}_{A} \mid \underbrace{\mathbf{x}_{t-1}}_{B}, \underbrace{\mathbf{u}_{0:t-1}, \mathbf{z}_{1:t-1}}_{C}) p(\underbrace{\mathbf{x}_{t-1}}_{B} \mid \underbrace{\mathbf{u}_{0:t-1}, \mathbf{z}_{1:t-1}}_{C}) \tag{13}$$

- putting in the result of Eq. 8, we highlight the transition model

$$= \sum_{\mathbf{x}_{t-1}} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_{t-1}) p(\mathbf{x}_{t-1} \mid \mathbf{u}_{0:t-1}, \mathbf{z}_{1:t-1}) \tag{14}$$

Filtering: Wrapup

- After our efforts, we figure out that the recursive filtering equation is the following:

$$p(\mathbf{x}_t \mid \mathbf{u}_{0:t-1}, \mathbf{z}_{1:t}) = \tag{15}$$

$$\eta_t p(\mathbf{z}_t \mid \mathbf{x}_t) \sum_{\mathbf{x}_{t-1}} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_{t-1}) p(\mathbf{x}_{t-1} \mid \mathbf{u}_{0:t-1}, \mathbf{z}_{1:t-1}) \tag{16}$$

- yet, if in the last term of the product in the summation, we would not have a dependancy from $\mathbf{u}_{t-1}$, we would have a recursive equation.
- luckily, $p(\mathbf{x}_{t-1} \mid \mathbf{u}_{0:t-1}, \mathbf{z}_{1:t-1}) = p(\mathbf{x}_{t-1} \mid \mathbf{u}_{0:t-2}, \mathbf{z}_{1:t-1})$, since the last control has no influence on $\mathbf{x}_{t-1}$ if we don't know $\mathbf{x}_t$.

Filtering: Wrapup

- We can finally write the recursive equation of filtering as:

$$\overbrace{p(\mathbf{x}_t \mid \mathbf{u}_{0:t-1}, \mathbf{z}_{1:t})}^{b(\mathbf{x}_t)} \tag{17}$$

$$= \eta_t p(\mathbf{z}_t \mid \mathbf{x}_t) \sum_{\mathbf{x}_{t-1}} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_{t-1}) \underbrace{p(\mathbf{x}_{t-1} \mid \mathbf{u}_{0:t-2}, \mathbf{z}_{1:t-1})}_{b(\mathbf{x}_{t-1})}$$

- during the estimation, we do not have the true distribution, but rather beliefs estimate.
- Equation 18 tells us how to update a current belief once new observations/controls become available

$$b(\mathbf{x}_t) = \eta_t p(\mathbf{z}_t \mid \mathbf{x}_t) \sum_{\mathbf{x}_{t-1}} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_{t-1}) b(\mathbf{x}_{t-1}) \tag{18}$$

Normalizer: η_t

- η_t is just a constant ensuring that $b(\mathbf{x}_t)$ is still a probability distribution.

$$\eta_t = \frac{1}{\sum_{\mathbf{x}_t} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_{t-1}) b(\mathbf{x}_{t-1})}$$

Filtering: Discrete case

```
float transition_model(int to, int from, int control);
float observation_model(int observarion, int state);

void filter(float* b, int n_states, int z, int u) {
  // clear the states
  float b_pred[n_states];
  for(int i=0; i<n_states; i++)
    b_pred[i]=0;

  // predict
  for(int i=0; i<n_states; i++)
    for(int j=0; j<n_states; j++)
      b_pred[i]+=transition_model(i,j,u)*b[j];

  // integrate the observation
  float inverse_eta = 0;
  for (int i=0; i<n_states; i++)
    inverse_eta += b_pred[i]*=observation_model(z,i);

  // normalize
  float eta = 1./inverse_eta;
  for (int i=0; i<n_states; i++)
    b[i] = b_pred[i] * eta;
}
```

Filtering: Alternative Formulation

Predict: incorporate in the last belief b_{t-1}, the most recent observation.

- From transition model and the last state, compute the following joint distribution through *chain rule*

$$p(\mathbf{x}_t, \mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1}) = p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_{t-1}) \underbrace{p(\mathbf{x}_t, \mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-2})}_{b_{t-1}} \tag{19}$$

- From the joint, remove $\mathbf{x}_{t-1}$ through *marginalization*

$$\underbrace{p(\mathbf{x}_t | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1})}_{b_{t|t-1}} = \sum_{\mathbf{x}_{t-1}} p(\mathbf{x}_t, \mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1}) \tag{20}$$

Filtering: Alternative Formulation

Update: from the predicted belief $b_{t|t-1}$, compute the joint distribution that predicts the observation.

- From transition model and the last state, compute the following joint distribution through *chain rule*

$$p(\mathbf{x}_t, \mathbf{z}_t | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1}) = p(\mathbf{z}_t | \mathbf{x}_t) p(\mathbf{x}_t, | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1}) \tag{21}$$

- Incorporate the current observation through *conditioning*, on the actual measurement

$$\underbrace{p(\mathbf{x}_t | z_{1:t}, u_{1:t-1})}_{b_{t|t}} = \frac{p(\mathbf{x}_t, \mathbf{z}_t | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1})}{p(\mathbf{z}_t | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1})} \tag{22}$$

Note: since we already know the value of $\mathbf{z}_t$, we do not need to compute the joint distribution for all possible values of $\mathbf{z} \in \mathcal{Z}$, but just for the current measurement.