Controller with Bayes Filter (`ctrlBF.m`)

Rowan McAllister

January 4, 2016

Contents

To control a dynamical system well we need to be able to estimate the current state of a system. Let us denote the latent system-state at time t as x_t. We'll also assume a simple observation model whereby our system's sensors periodically output a noisy version of the current system-state: $y_t = x_t + \epsilon$, where ϵ is a Gaussian white noise vector. A straightforward method of estimating x_t is thus to use the current sensor observation y_t, which can be inputted directly into our controller, depicted Fig. 1:

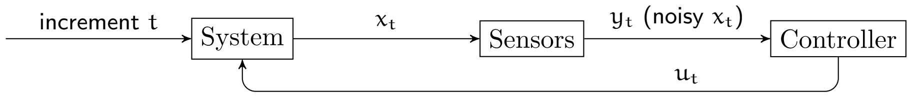


Figure 1: System using raw sensor signal for controller input.

The above solution may be adequate for low noise levels ϵ, but fail catastrophically otherwise. If large ϵ noise levels are directly injected into a controller configured with high gain parameters, the system will react wildly and may destabilise. To account for noisy sensors, we can *filter* the observation signal y_t before input into the controller, shown Fig. 2:

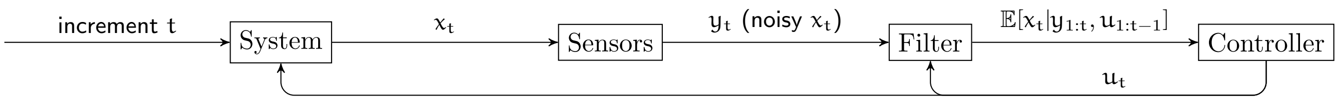


Figure 2: System with filtered controller input.

A Bayes filter (BF) maintains a belief-posterior on x, denoted B_{t+1}, conditioned on all information available thus far: the entire history of the system observations $y_{1:t}$ and applied control signals $u_{1:t-1}$. Conditioning on more information than the current observation y_t yields a more informed (thus accurate) estimate of x_t. Being a function of all observations, B_{t+1} is less susceptible to the noise injected into the most recent observation y_t, and consequently the controller's input is much smoother. To maintain B_{t+1} the BF makes two recursive updates per timestep:

1. Update step: Compute B_{tt} using prior belief $B_t = p(x_t)$ and observation likelihood $\mathcal{L}(x_t|y_t) = p(y_t|x_t)$,
2. Predict step: Compute B_{t+1} by mapping updated belief B_{tt} through transition model $p(x_{t+1}|x_t, u_t)$.

A directed graphical model of a Bayes filter is shown Fig. 3:

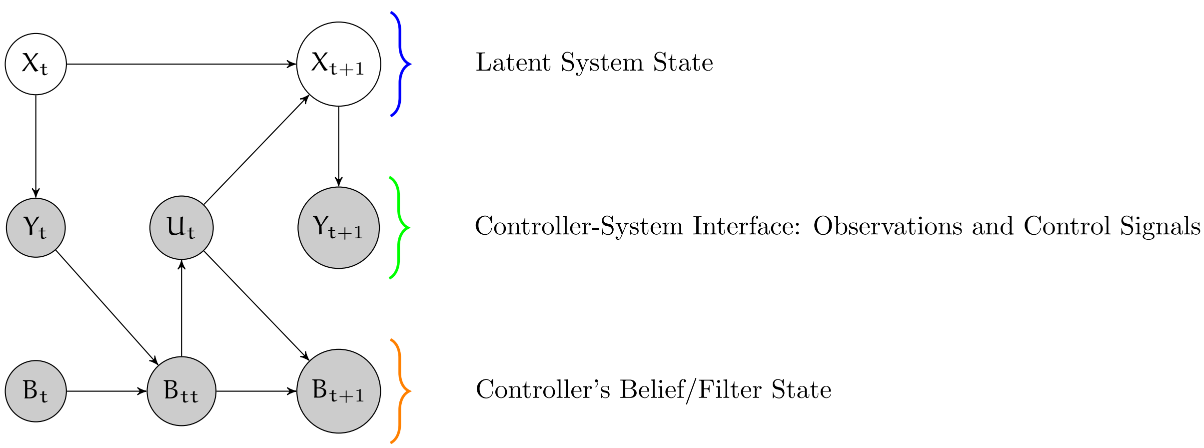


Figure 3: Directed graphical model of a system using a Bayes filter. A dynamical system begins in state X_t, which sensors observe noisily as Y_t. The observation Y_t is fused with the filter's prior on plausible system-states B_t, resulting in a posterior B_{tt}. The controller uses B_{tt} to decide control signal U_t. Finally, the control signal U_t is applied to the system, resulting in new state X_{t+1}, and also used by the BF to predict the system-state in the next time step B_{t+1}.

1 `ctrlBF.m` Inputs and Outputs

Terminology and Notation:

1. The subscripts:
   - $_t$ pertains to the current BF predicted state;
   - $_{tt}$ pertains to the current BF updated state w.r.t. noisy observation y_t;

- $_{t+1}$ pertains to the next BF prediction state w.r.t. dynamics model $\mathcal{T}$.

2. The circle superscript $^\circ$ denotes the *trigonometric* sin/cos angles of its operand.
3. Random variables and matrices are capitalised.

1.1 `ctrlBF.m` Inputs

A state-struct s_t is inputted with fields:

- m_{y_t} observation mean,
- S_{y_t} observation variance,
- m_{z_t} prior filter mean-of-mean,
- S_{z_t} prior filter variance-of-mean,
- `zc` $C_{x_t z_t}$: covariance of state and prior filter mean,
- V_t prior filter variance.

1.2 `ctrlBF.m` Local Variables

Block variables M, S and V are progressively expanded to (blue indicates computed with **`fillIn`** function, orange indicates only required when state representation is > 1 Markov grey indicates what was though to be required when state representation is > 1 Markov)), :

m_{x_t}
m_{z_t}
$m_{z_{tt}}$
$m_{z^\circ_{tt}}$
m_{u_t}
$m_{z_{t+1}}$

Block mean M

S_{x_t}	$C_{x_t z_t}$				
$C_{z_t x_t}$	S_{z_t}			$C_{z_t u_t}$	$C_{z_t z_{t+1}}$
		$S_{z_{tt}}$	$C_{z_{tt} z^\circ_{tt}}$	$C_{z_{tt} u_t}$	$C_{z_{tt} z_{t+1}}$
		$C_{z^\circ_{tt} z_{tt}}$	$S_{z^\circ_{tt}}$		
	$C_{u_t z_t}$	$C_{u_t z_{tt}}$		S_{u_t}	$C_{u_t z_{t+1}}$
	$C_{z_{t+1} z_t}$	$C_{z_{t+1} z_{tt}}$		$C_{z_{t+1} u_t}$	$S_{z_{t+1}}$

Block variance S

V_t
V_{tt}
V°_{tt}
V_{t+1}

Diagonal blocks of block variance V

1.3 `ctrlBF.m` Outputs

- M_{ctrl} control signal mean,
- S_{ctrl} control signal variance,
- C_{ctrl} $S_{x_t}^{-1}\mathbb{C}[X_t, U_t]$, inverse state variance times input-state output-control covariance (derived App. **??**),
- s_{t+1} state-struct with predicted filter fields:
  - m_{y_t} observation mean (unchanged),
  - S_{y_t} observation variance (unchanged),
  - $m_{z_{t+1}}$ predicted filter mean-of-mean (derived Table 2),
  - $S_{z_{t+1}}$ predicted filter variance-of-mean (derived Table 2),
  - `zc` $C_{x_{t+1}, z_{t+1}}$: covariance of state and predicted filter mean (derived App. J),
  - V_{t+1} predicted filter variance (derived Table 1, 2).

2 Rollouts with `ctrlBF.m`

We define a simulation of a system's possible evolution up to horizon H as a 'rollout'. As a system transitions from state x_t to x_{t+1}, the controller's Gaussian belief-distribution B_t will evolve w.r.t. simulated observations y_t and control signals u_t at each point in time t. We use lower case x and u to signify point masses, not distributions. A real-world test of system progresses as a rollout also (the only difference being observations are read from hardware, not simulated). A rollout proceeds as follows in Table 1:

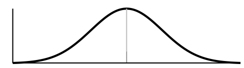
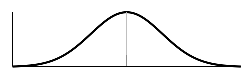
<table>
<tr><th>SYSTEM STATE</th><th></th><th>CONTROLLER'S BELIEF</th></tr>
<tr><td>$\downarrow x_t$</td><td></td><td>$\downarrow B_t$</td></tr>
<tr><td>Observation: sample y_t from $\mathcal{N}(x_t, S_\epsilon)$</td><td>$y_t$
$\rightarrow$</td><td>Updated Controller State: $B_{tt} \sim \mathcal{N}(z_{tt}, V_{tt})$

z_{tt} b_{tt}
$V_{tt} = (V_t^{-1} + S_\epsilon^{-1})^{-1}$
$z_{tt} = V_{tt}(V_t^{-1} z_t + S_\epsilon^{-1} y_t) = w_z z_t + w_y y_t$</td></tr>
<tr><td>$\downarrow x_t$</td><td></td><td>$\downarrow B_{tt}$</td></tr>
<tr><td>New Latent State: $x_{t+1} = \text{simulate}(x_t, u_t)$</td><td>$u_t$
$\leftarrow$</td><td>Controller signal: $u_t = \text{policy}(z_{tt}, z^\circ_{tt})$where</td></tr>
<tr><td>$\downarrow x_{t+1}$</td><td></td><td>$\downarrow B_{tt}, u_t$</td></tr>
<tr><td>$\downarrow x_{t+1}$</td><td></td><td>Predicted Controller State: $B_{t+1} \sim \mathcal{N}(z_{t+1}, V_{t+1})$

z_{t+1} b_{t+1}
m-code: $\{z_{t+1}, C, V_{t+1}\} =$
$\text{gph}\left(\mathcal{T}, \begin{bmatrix} z_{tt} \\ u_t \end{bmatrix}, V_{tt}\right)$</td></tr>
<tr><td>$\downarrow x_{t+1}$</td><td></td><td>$\downarrow B_{t+1}$</td></tr>
</table>

Table 1: Rollout Flowchart.

3 Propagate with `ctrlBF.m`

Training our controller requires computing the value function: the expected loss over possible futures. Unfortunately a rollout will only simulate how one possible future may be reached. Instead, we consider a continuum of possible futures as a distribution on the latent states X_t. The initial latent state X_0 may be a point mass, but after one timestep X_1 will be a distribution owing to inherent system stochasticity (or 'process noise'). As our belief defines a belief-distribution over each plausible latent state, our belief $B_t \sim \mathcal{N}(z_t, V_t)$ is thus redefined as a distribution over distributions. We define a different belief distributions using a distribution on belief mean parameter: $Z_t = \mathcal{N}(m_{z_t}, S_{z_t})$. We make the simplifying approximation that V_t is constant across each latent state X_t. Thus, our Gaussian belief distribution with an uncertain mean parameter is therefore a hierarchical Gaussian distribution function: $B_t = \mathcal{N}(Z_t, V_t) = \mathcal{N}(\mathcal{N}(m_{z_t}, S_{z_t}), V_t)$. We define the evolution of this richer belief function over time as a 'propagation', seen Table 2:

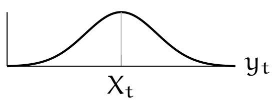
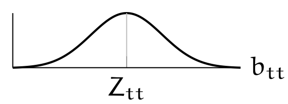
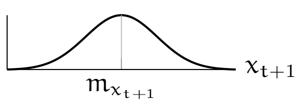
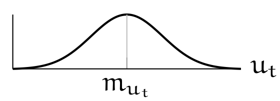
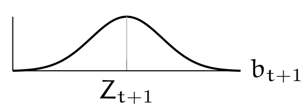
System States		Controller's Beliefs
$\downarrow X_t \sim \mathcal{N}(m_{x_t}, S_{x_t})$		$\downarrow B_t$
Observation: $Y_t \sim \mathcal{N}(\mathcal{N}(m_{x_t}, S_{x_t}), S_\epsilon)$  $Y_t \sim \mathcal{N}(m_{y_t}, S_{y_t})$ $m_{y_t} = \mathbb{E}[Y_t] = m_{x_t}$ $S_{y_t} = \mathbb{V}[Y_t] = S_{x_t} + S_\epsilon$	Y_t $\rightarrow$	Updated Controller State: $B_{tt} \sim \mathcal{N}(\mathcal{N}(m_{z_{tt}}, S_{z_{tt}}), V_{tt})$  $V_{tt} = (V_t^{-1} + S_\epsilon^{-1})^{-1}$ $Z_{tt} \sim \mathcal{N}(m_{z_{tt}}, S_{z_{tt}})$ $Z_{tt} = V_{tt}(V_t^{-1} Z_t + S_\epsilon^{-1} Y_t) = w_z Z_t + w_y Y_t$ $m_{z_{tt}} = w_z m_{z_t} + w_y m_{y_t}$ $S_{z_{tt}} = [w_z, w_y] \begin{bmatrix} S_{z_t} & C_{z_t x_t} \\ C_{x_t z_t} & S_{y_t} \end{bmatrix} \begin{bmatrix} w_z^\top \\ w_y^\top \end{bmatrix}$ $[C_{z_{tt} z_t}, C_{z_{tt} x_t}] = [w_z, w_y] \begin{bmatrix} S_{z_t} & C_{z_t x_t} \\ C_{x_t z_t} & S_{x_t} \end{bmatrix}$
$\downarrow X_t$		$\downarrow B_{tt}$
New Latent State: $X_{t+1} \sim \mathcal{N}(m_{x_{t+1}}, S_{x_{t+1}})$  m-code: $\{m_{x_{t+1}}, S_{x_{t+1}}, C\} =$ $\text{gph}\left(\mathcal{T}, \begin{bmatrix} m_{x_t} \\ m_{u_t} \end{bmatrix}, \begin{bmatrix} S_{x_t} & C_{x_t u_t} \\ C_{u_t x_t} & S_{u_t} \end{bmatrix}\right)$	U_t $\leftarrow$	Controller signal: $U_t \sim \mathcal{N}(m_{u_t}, S_{u_t})$  Policy $\pi: \begin{bmatrix} m_{z_{tt}} \\ m_{z_{tt}^\circ} \end{bmatrix} \times \begin{bmatrix} S_{z_{tt}} & C_{z_{tt} z_{tt}^\circ} \\ C_{z_{tt}^\circ z_{tt}} & S_{z_{tt}^\circ} \end{bmatrix} \rightarrow m_{u_t} \times S_{u_t}$
$\downarrow X_{t+1}$		$\downarrow B_{tt}$, U_t
$\downarrow X_{t+1}$		Predicted Controller State: $B_{t+1} \sim \mathcal{N}(\mathcal{N}(m_{z_{t+1}}, S_{z_{t+1}}), V_{t+1})$  $Z_{t+1} \sim \mathcal{N}(m_{z_{t+1}}, S_{z_{t+1}})$ m-code: $\{m_{z_{t+1}}, S_{z_{t+1}}, C, V_{t+1}\} =$ $\text{gph}\left(\mathcal{T}, \begin{bmatrix} m_{z_{tt}} \\ m_{u_t} \end{bmatrix}, \begin{bmatrix} S_{z_{tt}} & C_{z_{tt}, u_t} \\ C_{u_t, z_{tt}} & S_{u_t} \end{bmatrix}, V_{tt}\right)$
$\downarrow X_{t+1}$		$\downarrow B_{t+1}$

Table 2: Propagate Flowchart.

3.1 Comments:

1. Although future values of Y are currently unknown, we already know our strength-of-belief in Y will be S_ϵ.
2. Note the controller outputs real numbers, not distributions. However, a distribution over future control signals exists since we are currently uncertain about what the value of Z will be. We also (for now) assume the controller is a function of the belief-mean Z only, and not also a function of strength-of-belief V.

3. We can express the joint probability of all uncertain and hierarchically-uncertain terms involved:

$$\begin{bmatrix} Z_t \\ B_t \\ X_t \\ Y_t \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m_{z_t} \\ m_{z_t} \\ m_{x_t} \\ m_{x_t} \end{bmatrix}, \begin{bmatrix} S_{z_t} & S_{z_t} & C_{x_t z_t}^\top & C_{x_t z_t}^\top \\ S_{z_t} & S_{z_t} + V_t & C_{x_t z_t}^\top & C_{x_t z_t}^\top \\ C_{x_t z_t} & C_{x_t z_t} & S_{x_t} & S_{x_t} \\ C_{x_t z_t} & C_{x_t z_t} & S_{x_t} & S_{x_t} + S_\epsilon \end{bmatrix}\right).$$

4. `ctrlBF.m` instead uses rearranged expressions for V_{tt} and $m_{z_{tt}}$:

   (a) $V_{tt} = S_\epsilon(V_t + S_\epsilon)^{-1}V_t = V_t(V_t + S_\epsilon)^{-1}S_\epsilon$,
   (b) $m_{z_{tt}} = S_\epsilon(V_t + S_\epsilon)^{-1}m_{z_t} + V_t(V_t + S_\epsilon)^{-1}m_{y_t}$.

3.2 Pseudocode

Let s represent the system state, s° the augmented system state, b the filter state, b° the augmented filter state, u the control signal (either point masses, distributions, or hierarchical distributions). A particular parameterisation of the controller is evaluated with:

total cost up to horizon, $\frac{\text{d total cost up to horizon}}{\text{d}p}$ ← `value`(s, @`ctrlBF` parameterised by p, @`dynmodel`, @`cost`)

FOR time = now to horizon:

1. $s_{\text{predicted}}$, $\frac{\partial s_{\text{predicted}}}{\partial\{s,p\}}$ ← `propagate`(s, b)
   (a) initialise $\frac{\partial s_{\text{predicted}}}{\partial\{s,p\}}$
   (b) u, $\frac{\partial s_{\text{predicted}}}{\partial\{s,p\}}$ ← `ctrlBF`(b, s, $\frac{\partial s_{\text{predicted}}}{\partial\{s,p\}}$)
      i. b_{updated} ← `update`(b, s noisily observed)
      ii. b°_{updated} ← `gTrig`(b_{updated})
      iii. u ← `ctrlBF.policy`(b_{updated})
      iv. $b_{\text{predicted}}$ ← `ctrlBF.dynmodel`(b_{updated}, b°_{updated}, u)
      v. contribute to $\frac{\partial s_{\text{predicted}}}{\partial\{s,p\}}$
   (c) s° ← `gTrig`(s)
   (d) $s_{\text{predicted}}$ ← `dynmodel`(s, s°, u)
   (e) contribute to $\frac{\partial s_{\text{predicted}}}{\partial\{s,p\}}$
2. update $\frac{\text{d}s_{\text{predicted}}}{\text{d}p}$
3. cost, $\frac{\partial \text{cost}}{\partial s_{\text{predicted}}}$ ← `cost`($s_{\text{predicted}}$)
4. update total cost so far and $\frac{\text{d total cost so far}}{\text{d}p}$
5. s ← $s_{\text{predicted}}$

ENDFOR

The controller is then trained using policy search: minimising total cost by descending the $\frac{\text{d total cost up to horizon}}{\text{d}p}$ gradient.

A Derivation of $\mathbb{C}[M_t, Z_{tt}]$:

Let $M_t \doteq \begin{bmatrix} X_t \\ Z_t \end{bmatrix}$, and $C_{xz} \doteq \mathbb{C}[X_t, Z_t]$, then

$$
\begin{aligned}
\mathbb{C}[M_t, Z_{tt}] = \begin{bmatrix} \mathbb{C}[X_t, Z_{tt}] \\ \mathbb{C}[Z_t, Z_{tt}] \end{bmatrix} &= \begin{bmatrix} \mathbb{C}[X_t, w_y Y_t + w_z Z_t] \\ \mathbb{C}[Z_t, w_y Y_t + w_z Z_t] \end{bmatrix} \\
&= \begin{bmatrix} S_{X_t} w_y^\top + C_{xz} w_z^\top \\ C_{zx} w_y^\top + S_{Z_t} w_z^\top \end{bmatrix} \\
&= \underbrace{\begin{bmatrix} S_{X_t} + C_{xz} \\ C_{zx} + S_{Z_t} \end{bmatrix}}_{S_{M_t}} \underbrace{[w_y, w_z]^\top}_{w^\top} \\
&= S_{M_t} w^\top
\end{aligned} \tag{1}
$$

B Derivation of $\mathbb{C}[Z_{tt}, U_t]$:

We begin with:

$$
\begin{aligned}
\mathbb{C}[\{Z_{tt}; Z_{tt}^\circ\}, U_t] &= S_{\{Z_{tt}, Z_{tt}^\circ\}} \underbrace{C_{poli}[\{Z_{tt}, Z_{tt}^\circ\}, U_t]}_{p} \\
\begin{bmatrix} \mathbb{C}[Z_{tt}, U_t] \\ \mathbb{C}[Z_{tt}^\circ, U_t] \end{bmatrix} &= \begin{bmatrix} S_{Z_{tt}} & \mathbb{C}[Z_{tt}, Z_{tt}^\circ] \\ \mathbb{C}[Z_{tt}^\circ, Z_{tt}] & S_{Z_{tt}^\circ} \end{bmatrix} p \\
\therefore \mathbb{C}[Z_{tt}, U_t] &= [S_{Z_{tt}}, \mathbb{C}[Z_{tt}, Z_{tt}^\circ]] p \\
&= S_{Z_{tt}} \underbrace{[I, C_{gtrig}[Z_{tt}, Z_{tt}^\circ]]}_{g} p \\
&= S_{Z_{tt}} g p
\end{aligned} \tag{2}
$$

C Derivation of $\mathbb{C}[M_t, U_t]$:

$$
\begin{aligned}
\mathbb{C}[M_t, U_t] &= \mathbb{C}[M_t, Z_{tt}] S_{Z_{tt}}^{-1} \mathbb{C}[Z_{tt}, U_t] \qquad \text{(TODO: prove)} \\
&= S_{M_t} w^\top g p
\end{aligned}
$$

`ctrlNF.m` case:

$$
\begin{aligned}
\mathbb{C}[\{Y_t, Y_t^\circ\}, U_t] &= S_{\{Y_t, Y_t^\circ\}} C_{poli}[\{Y_t, Y_t^\circ\}, U_t] \\
\begin{bmatrix} \mathbb{C}[Y_t, U_t] \\ \mathbb{C}[Y_t^\circ, U_t] \end{bmatrix} &= \begin{bmatrix} S_{Y_t} & \mathbb{C}[Y_t, Y_t^\circ] \\ \mathbb{C}[Y_t^\circ, Y_t] & S_{Y_t^\circ} \end{bmatrix} C_{poli}[\{Y_t, Y_t^\circ\}, U_t] \\
\therefore \mathbb{C}[Y_t, U_t] &= [S_{Y_t}, \mathbb{C}[Y_t, Y_t^\circ]] C_{poli}[\{Y_t, Y_t^\circ\}, U_t] \\
&= S_{Y_t} [I, C_{gtrig}[Y_t, Y_t^\circ]] C_{poli}[\{Y_t, Y_t^\circ\}, U_t]
\end{aligned}
$$

And as a set of weights:

$$
\mathbb{C}[X_t, U_t] = S_{X_t} [I, C_{gtrig}[Y_t, Y_t^\circ]] C_{poli}[\{Y_t, Y_t^\circ\}, U_t]
$$

D Derivation of $\mathbb{C}[Z_{tt}, Z_{t+1}]$:

$$\begin{aligned}
\mathbb{C}[\{Z_{tt}, U_t\}, Z_{t+1}] &= S_{\{Z_{tt}, U_t\}} \underbrace{C_{dyn}[\{Z_{tt}, U_t\}, Z_{t+1}]}_{d_z} \\
\begin{bmatrix} \mathbb{C}[Z_{tt}, Z_{t+1}] \\ \mathbb{C}[U_t, Z_{t+1}] \end{bmatrix} &= \begin{bmatrix} S_{Z_{tt}} & \mathbb{C}[Z_{tt}, U_t] \\ \mathbb{C}[U_t, Z_{tt}] & S_{U_t} \end{bmatrix} d_z \\
\therefore \mathbb{C}[Z_{tt}, Z_{t+1}] &= [S_{Z_{tt}}, \mathbb{C}[Z_{tt}, U_t]] d_z \\
&= S_{Z_{tt}} [I, gp] d_z \qquad \text{using Eq. 2}
\end{aligned} \tag{3}$$

E Derivation of $\mathbb{C}[M_t, Z_{t+1}]$:

$$\begin{aligned}
\mathbb{C}[M_t, Z_{t+1}] &= \mathbb{C}[M_t, Z_{tt}] S_{Z_{tt}}^{-1} \mathbb{C}[Z_{tt}, Z_{t+1}] \qquad (\text{TODO: prove}) \\
&= S_{M_t} w^\top [I, gp] d_z
\end{aligned} \tag{4}$$

F Derivation of `ctrlBF.m`'s Output C_{ctrlbf}:

$$\begin{aligned}
C_{ctrlbf} &\doteq S_{M_t}^{-1} \mathbb{C}[M_t, \{U_t; Z_{t+1}\}] \\
&= S_{M_t}^{-1} \left[\mathbb{C}[M_t, U_t], \quad \mathbb{C}[M_t, Z_{t+1}] \right] \\
&= \left[w^\top gp, \quad w^\top [I, gp] d_z \right]
\end{aligned} \tag{5}$$

G Derivation of $\mathbb{C}[M_t, X_{t+1}]$:

$$\begin{aligned}
\mathbb{C}[M_t, X_{t+1}] &= \mathbb{C}[M_t, \{X_t; U_t\} \mathbb{V}[\{X_t; U_t\}]^{-1} \mathbb{C}[\{X_t; U_t\}, X_{t+1}] \\
&= \left[\mathbb{C}[M_t, X_t], \quad \mathbb{C}[M_t, U_t] \right] \underbrace{C_{propdyn}[\{X_t; U_t\}, X_{t+1}]}_{d_x} \\
&= S_{M_t} \left[\begin{bmatrix} I \\ 0 \end{bmatrix}, \quad w^\top gp \right] d_x
\end{aligned} \tag{6}$$

H Derivation of `propagate.m`'s Output C_{prop}:

Let $M_{t+1} \doteq \begin{bmatrix} X_{t+1} \\ Z_{t+1} \end{bmatrix}$. Goal is to compute:

$$C_{prop} \doteq S_{M_t}^{-1} \mathbb{C}[M_t, M_{t+1}]$$

H.1 `propagate.m` with ctrlBF

By combining Eq 4 and Eq 6 we have:

$$\begin{aligned}
C_{prop}^{BF} &= S_{M_t}^{-1} \mathbb{C}[M_t, \{X_{t+1}; Z_{t+1}\}] \\
&= S_{M_t}^{-1} \left[\mathbb{C}[M_t, X_{t+1}], \quad \mathbb{C}[M_t, Z_{t+1}] \right] \\
&= \left[\left[\begin{bmatrix} I \\ 0 \end{bmatrix}, \quad w^\top gp \right] d_x, \quad w^\top [I, gp] d_z \right] \\
&= \left[\left[\texttt{eye(F,D)}, \quad \texttt{C_ctrlbf(1:U)} \right] \texttt{d_x}, \quad \texttt{C_ctrlbf(U+1:end)} \right]
\end{aligned}$$

H.2 `propagate.m` with ctrlNF

$$\begin{aligned}
C_{prop}^{NF} &= S_{X_t}^{-1} \mathbb{C}[X_t, X_{t+1}] \\
&= \left[I, \quad gp \right] d_x \\
&= [I, C_{ctrlnf}] d_x
\end{aligned}$$

H.3 `propagate.m` with ctrlBF, and exact $\mathbb{C}[M_t, M_{t+1}]$

$$\begin{aligned}
C^{BF}_{\mathtt{propExact}} &= S^{-1}_{M_t}\mathbb{C}[M_t, M_{t+1}] \\
&= S^{-1}_{M_t}\mathbb{C}[M_t, \{X_t; Z_{tt}; U_t\}]\underbrace{\mathbb{V}[X_t; Z_{tt}; U_t]^{-1}\mathbb{C}[[X_t; Z_{tt}; U_t], M_{t+1}]}_{C_{\mathtt{gphJoint}}} \\
&= \left[\begin{bmatrix} I \\ 0 \end{bmatrix}, w^\top, w^\top gp\right] C_{\mathtt{gphJoint}}
\end{aligned}$$

I Derivation of $\mathbb{C}[X_{t+1}, Z_{t+1}]$:

Using the top half of Eq. 4 to express $\mathbb{C}[X_t, Z_{t+1}]$, and part of Eq. 3 to express $\mathbb{C}[U_t, Z_{t+1}]$, we have:

$$\begin{aligned}
\mathbb{C}[X_{t+1}, Z_{t+1}] &= \underbrace{\mathbb{C}[X_{t+1}, \{X_t; U_t\}]\mathbb{V}[\{X_t; U_t\}]^{-1}}_{d_x^\top}\mathbb{C}[\{X_t; U_t\}, Z_{t+1}] \\
&= d_x^\top \begin{bmatrix} \mathbb{C}[X_t, Z_{t+1}] \\ \mathbb{C}[U_t, Z_{t+1}] \end{bmatrix} \\
&= d_x^\top \begin{bmatrix} [S_{X_t}, \mathbb{C}[X_t, Z_t]]w^\top [I, gp] d_z \\ [\mathbb{C}[U_t, Z_{tt}],\ S_{U_t}] d_z \end{bmatrix} \\
&= d_x^\top \begin{bmatrix} [S_{X_t}, C_{xz}]w^\top [I, gp] \\ [p^\top g^\top S_{Z_{tt}},\ S_{U_t}] \end{bmatrix} d_z
\end{aligned}$$

J Derivation of Exact $\mathbb{C}[X_{t+1}; Z_{t+1}]$:

Goal is (with help from propagate.m's computation of X_{t+1}) to compute:

$$s_{t+1}.\mathtt{zc} \triangleq \mathbb{C}[X_{t+1}, Z_{t+1}].$$

We simplify our graphical model (Fig. 3) such that X_{t+1} and B_{t+1} are the output of two $\mathcal{GP}$s with a common input, seen Fig.5:

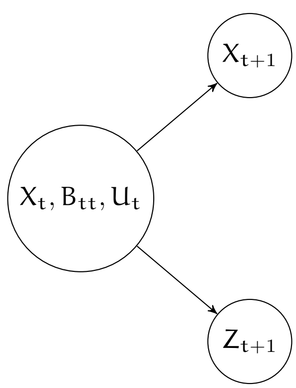


Figure 5: Simplified directed graphical model of Fig. 3.

The only tricky bit is X_{t+1} and Z_{t+1} use different (yet overlapping) subsets of the now common input $\{X_t, B_{tt}, U_t\}$. The common input's joint expression is given below, where X_{t+1}'s subset Σ_x is blue, Z_{t+1}'subset Σ_z and V_z is red, and where they overlap Σ_{xz} is purple.

$$\underbrace{\begin{bmatrix} X_t \\ B_{tt} \\ U_t \end{bmatrix}}_{y} \sim \mathcal{N}\left(\mathcal{N}\left(\underbrace{\begin{bmatrix} m_{x_t} \\ m_{z_{tt}} \\ m_{u_t} \end{bmatrix}}_{m}, \underbrace{\begin{bmatrix} S_{x_t} & C_{x_t,z_{tt}} & C_{x_t,u_t} \\ C_{z_{tt},x_t} & S_{z_{tt}} & C_{z_{tt},u_t} \\ C_{u_t,x_t} & C_{u_t,z_{tt}} & S_{u_t} \end{bmatrix}}_{\Sigma}\right), \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & V_{tt} & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{V}\right). \tag{7}$$

To re-express Λ_x to match the size of Σ (Eq. 7) whilst encoding the fact that X_{t+1} is conditionally independent of B_{tt} given X_t and U_t, we set the new diagonal elements corresponding to Z_{tt} as ∞:

$$\begin{aligned}
\Lambda_x = \mathrm{diag}(\lambda_x, \lambda_u) \quad &\rightarrow \quad \hat{\Lambda}_x = \mathrm{diag}(\lambda_x, \infty, \lambda_u), \\
\Lambda_z = \mathrm{diag}(\lambda_z, \lambda_u) \quad &\rightarrow \quad \hat{\Lambda}_z = \mathrm{diag}(\infty, \lambda_z, \lambda_u).
\end{aligned}$$

Now to compute the covariance of $\mathcal{GP}_x$'s output and $\mathcal{GP}_z$'s output given the common uncertain input (m, Σ) of both $\mathcal{GP}$s, we use the following identity from gph.pdf (noting that the inverse of a matrix A whose element $A_{ij} = \infty$, is s.t. $(A^{-1})_{kl} = 0$ if $k = j$ or $l = i$, otherwise populated by values of submatrix $(M_{I\setminus\{i\},J\setminus\{j\}})^{-1}$):

$$\mathbb{C}[X_{t+1}, Z_{t+1}] \quad = \quad \mathbb{C}[f_x^*, f_z^* | \mathbf{m}, \Sigma] \;=\; s_x^2 s_z^2 \big[\beta_x^\top (Q^{xz} - q^x q^{z\top})\beta_z\big] + C_x^\top \Sigma \theta_z + \theta_x^\top \Sigma C_z + \theta_x^\top \Sigma \theta_z,$$

where

$$\begin{aligned}
q_i^x &= q(y_i, \mathbf{m}, \hat{\Lambda}_x, \Sigma + V), \\
&= |\hat{\Lambda}_x^{-1}(\Sigma + V) + I|^{-1/2} \exp\big(-\tfrac{1}{2}(y_i - \mathbf{m})[\hat{\Lambda}_x + \Sigma + V]^{-1}(y_i - \mathbf{m})\big), \\
&= |\Lambda_x^{-1}(\Sigma_x + V_x) + I|^{-1/2} \exp\big(-\tfrac{1}{2}(x_i - \mathbf{m}_x)[\Lambda_x + \Sigma_x + V_x]^{-1}(x_i - \mathbf{m}_x)\big), \\
&= q(x_i, \mathbf{m}_x, \Lambda_x, \Sigma_x + V_x), \\
q_i^z &= q(y_i, \mathbf{m}, \hat{\Lambda}_z, \Sigma + V), \\
&= q(z_i, \mathbf{m}_z, \Lambda_z, \Sigma_z + V_z), \\
Q_{ij}^{xz} &= Q\big(y_i, y_j, \hat{\Lambda}_x, \hat{\Lambda}_z, V, \mathbf{m}, \Sigma\big), \\
&= c_2\, q(y_i, m, \Lambda_x, V_x)\, q(y_j, m, \Lambda_z, V_z) \exp\big(\tfrac{1}{2}\mathbf{r}^\top \big[(\hat{\Lambda}_x + V)^{-1} + (\hat{\Lambda}_z + V)^{-1} + \Sigma^{-1}\big]^{-1}\mathbf{r}\big), \\
\mathbf{r} &= (\hat{\Lambda}_x + V)^{-1}(y_i - m) + (\hat{\Lambda}_z + V)^{-1}(y_j - m), \\
c_2 &= \big|\big((\hat{\Lambda}_x + V)^{-1} + (\hat{\Lambda}_z + V)^{-1}\big)\Sigma + I\big|^{-1/2}.
\end{aligned}$$

We see the extended dimensionality has had no effect on the values on q^x and q^z (size $n \times E$). The value of Q^{xz} (size $E \times E$), however, is dependent on each element of the Σ matrix, thus we must compute the full Σ.

Old Actions

Let $(\cdot)^p$ signify the predicted state variables, and $(\cdot)^r$ signify the rest of the state (e.g. subset of the previous state if the state representation is > 1 order Markov, $(X_t)_{E+U:D}$, and the previous action U_t) E.g. $X_{t+1} = [X_{t+1}^r; X_{t+1}^p] = [(X_t)_{E+U:D}; U_t; X_{t+1}^p]$. The joint of X_{t+1} and Z_{t+1} is given in Fig. 6:

$$\mathbb{V}\begin{bmatrix} X_{t+1} \\ Z_{t+1} \end{bmatrix} = \mathbb{V}\begin{bmatrix} X_{t+1}^r \\ X_{t+1}^p \\ Z_{t+1}^r \\ Z_{t+1}^p \end{bmatrix} \quad = \quad \begin{bmatrix} \mathbb{V}[X_{t+1}^r] & \mathbb{C}[X_{t+1}^r, X_{t+1}^p] & \mathbb{C}[X_{t+1}^r, Z_{t+1}^r] & \mathbb{C}[X_{t+1}^r, Z_{t+1}^p] \\ \mathbb{C}[X_{t+1}^p, X_{t+1}^r] & \mathbb{V}[X_{t+1}^p] & \mathbb{C}[X_{t+1}^p, Z_{t+1}^r] & \mathbb{C}[X_{t+1}^p, Z_{t+1}^p] \\ \mathbb{C}[Z_{t+1}^r, X_{t+1}^r] & \mathbb{C}[Z_{t+1}^r, X_{t+1}^p] & \mathbb{V}[Z_{t+1}^r] & \mathbb{C}[Z_{t+1}^r, Z_{t+1}^p] \\ \mathbb{C}[Z_{t+1}^p, X_{t+1}^r] & \mathbb{C}[Z_{t+1}^p, X_{t+1}^p] & \mathbb{C}[Z_{t+1}^p, Z_{t+1}^r] & \mathbb{V}[Z_{t+1}^p] \end{bmatrix}.$$

Figure 6: We compute $\mathbb{C}[X_{t+1}^p, Z_{t+1}^p]$ using `gph.m`. The full $\mathbb{C}[X_{t+1}, Z_{t+1}]$ is composed of the blue and red members.

Linear Approximations

$\mathbb{C}[X_{t+1}, Z_{t+1}]$:

Computing $\mathbb{C}[X_{t+1}, Z_{t+1}]$ is an expensive bottleneck for `ctrlBF.m`. We can instead approximate $\mathbb{C}[X_{t+1}, Z_{t+1}]$, by assuming that both X_{t+1}, and Z_{t+1}, are linear functions of $[X_t; B_{tt}; U_t]$. Both predictions use a dynamics model each, which outputs covariance information with implicit premultiplied-input-variance-inverses, C_{dynx} and C_{dynz}, seen in Fig. 7. We can treat C_{dynx} and C_{dynz} as fixed weights, i.e. $X_{t+1} \approx C_{dynx}^\top [X_t; B_{tt}; U_t]$, and $Z_{t+1} \approx C_{dynz}^\top [X_t; B_{tt}; U_t]$. Thus $\mathbb{C}[X_{t+1}, Z_{t+1}] \approx C_{dynx}^\top \mathbb{V}[[X_t; B_{tt}; U_t]] C_{dynz}$.

$\mathbb{V}[X_{t+1}]^{-1}\mathbb{C}[X_{t+1}, Z_{t+1}]$:

Alternatively, we might be interested in approximating the covariance with the following implicit inverse $\mathbb{V}[X_{t+1}]^{-1}\mathbb{C}[X_{t+1}, Z_{t+1}]$. We can use the above linear approximations. Let us decompose PSD $\Sigma = V^\top D V = V^\top D^{\frac{1}{2}} D^{\frac{1}{2}} V = L^\top L$. And define $A = C_{dynx}^\top L^\top$. We have:

$$\begin{aligned}
\mathbb{V}[X_{t+1}]^{-1}\mathbb{C}[X_{t+1}, Z_{t+1}] &\approx \mathbb{V}[C_{dynx}^\top [X_t; B_{tt}; U_t]]^{-1} C_{dynx}^\top \mathbb{V}[[X_t; B_{tt}; U_t]] C_{dynz} \\
&= (C_{dynx}^\top \Sigma C_{dynx})^{-1} C_{dynx}^\top \Sigma C_{dynz} \\
&= (C_{dynx}^\top L^\top L C_{dynx})^{-1} C_{dynx}^\top L^\top L C_{dynz} \\
&= (A A^\top)^{-1} A L C_{dynz} \\
&= A (A^\top A)^{-1} L C_{dynz} \\
&= C_{dynx}^\top L^\top (L C_{dynx} C_{dynx}^\top L^\top)^{-1} L C_{dynz} \\
&= C_{dynx}^\top (C_{dynx} C_{dynx}^\top)^{-1} C_{dynz}
\end{aligned}$$

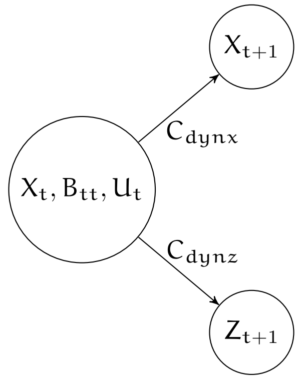


Figure 7: Simplified directed graphical model of Fig. 3 with linear weights.

K Joint Representation

Here we compute the joint expression of all uncertain terms involved in ctrlBF. We can think of the belief distribution B over state X, as a hierarchical representation: $B \sim \mathcal{N}(Z, V)$, where the mean parameter, $Z \sim \mathcal{N}(m_z, S_z)$, is itself uncertain. Equivalently, we might consider B to be the sum of two (initially) independent random variables: $B = Z + Q$, where $Q \sim \mathcal{N}(0, V)$.

Firstly, given:

$$\begin{aligned} p(z_t) &\sim \mathcal{N}(m_{z_t}, S_{z_t}) \\ p(b_t|z_t) &\sim \mathcal{N}(z_t, V_t) \\ p(b_t) &\sim \mathcal{N}(m_{z_t}, S_{b_t}) \\ S_{b_t} &= S_{z_t} + V_t \\ p(y_t|b_t) &\sim \mathcal{N}(b_t, S_\epsilon) \end{aligned}$$

then

$$\begin{aligned} p(b_t|y_t, z_t) &\propto p(y_t|b_t, z_t)p(b_t|z_t) \\ &= p(y_t|b_t)p(b_t|z_t) \\ &\sim \mathcal{N}\Big(w_z z_t + w_y y_t, \ \ (S_\epsilon^{-1} + V_t^{-1})^{-1}\Big) \\ w_z &= S_\epsilon(S_\epsilon + V_t)^{-1} \\ w_y &= V_t(S_\epsilon + V_t)^{-1} \\ p(b_t|y_t) &\propto p(y_t|b_t)p(b_t) \\ &\sim \mathcal{N}\Big(S_{b|y}(S_\epsilon^{-1} y_t + S_{b_t}^{-1} m_{z_t}), \ \ S_{b|y}\Big) \\ S_{b|y} &= (S_\epsilon^{-1} + S_{b_t}^{-1})^{-1} \end{aligned}$$

Now consider the following joint belief expressions:

$$\begin{aligned}
p\begin{pmatrix} b_t \\ z_t \end{pmatrix} &\sim \left(\begin{bmatrix} m_{z_t} \\ m_{z_t} \end{bmatrix}, \begin{bmatrix} S_{b_t} & S_{z_t} \\ S_{z_t} & S_{z_t} \end{bmatrix} \right) \\
\to p(b_{tt}, z_{tt}) \triangleq p(\{b_t, z_t\}|y_t) &\propto p(y_t|b_t, z_t)p(b_t, z_t) \\
&= p(y_t|b_t)p(b_t, z_t) \\
&\sim \mathcal{N}\Big(\Sigma \begin{bmatrix} S_\epsilon & 0 \\ 0 & \infty \end{bmatrix}^{-1} \begin{bmatrix} y_t \\ 0 \end{bmatrix} + \Sigma \begin{bmatrix} S_{b_t} & S_{z_t} \\ S_{z_t} & S_{z_t} \end{bmatrix}^{-1} \begin{bmatrix} m_{z_t} \\ m_{z_t} \end{bmatrix}, \ \Sigma \Big) \\
\Sigma^{-1} &= \begin{bmatrix} S_\epsilon^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} S_{b_t} & S_{z_t} \\ S_{z_t} & S_{z_t} \end{bmatrix}^{-1} \\
&= \begin{bmatrix} S_\epsilon^{-1} + (S_{b_t} - S_{z_t})^{-1} & -S_{b_t}^{-1} S_{z_t} (S_{z_t} - S_{z_t} S_{b_t}^{-1} S_{z_t})^{-1} \\ -(S_{b_t} - S_{z_t})^{-1} & (S_{z_t} - S_{z_t} S_{b_t}^{-1} S_{z_t})^{-1} \end{bmatrix} \\
\Sigma &= \begin{bmatrix} S_{b_t}(S_{b_t} + S_\epsilon)^{-1} S_\epsilon & S_\epsilon (S_{b_t} + S_\epsilon)^{-1} S_{z_t} \\ S_{z_t}(S_{b_t} + S_\epsilon)^{-1} S_\epsilon & S_{z_t} - S_{z_t}(S_{b_t} + S_\epsilon)^{-1} S_{z_t} \end{bmatrix} \\
&= \begin{bmatrix} S_{b|y} & C_{bz|y} \\ C_{bz|y}^\top & S_{z|y} \end{bmatrix} \\
C_{bz|y} &= S_\epsilon (S_{b_t} + S_\epsilon)^{-1} S_{z_t} = S_{b|y} S_{b_t}^{-1} S_{z_t} \\
S_{z|y} &= S_{z_t} - S_{z_t}(S_{b_t} + S_\epsilon)^{-1} S_{z_t} = (V_t + S_\epsilon)(S_{b_t} + S_\epsilon)^{-1} S_{z_t} = (S_{z_t}^{-1} + (V_t + S_\epsilon)^{-1})^{-1} \\
\therefore p(b_t, z_t|y_t) &\sim \mathcal{N}\Big(\begin{bmatrix} S_{b|y} & C_{bz|y} \\ C_{bz|y}^\top & S_{z|y} \end{bmatrix} \begin{bmatrix} S_\epsilon^{-1} y_t \\ 0 \end{bmatrix} + \begin{bmatrix} S_\epsilon (S_{b_t} + S_\epsilon)^{-1} & 0 \\ -S_{z_t}(S_{b_t} + S_\epsilon)^{-1} & I \end{bmatrix} \begin{bmatrix} m_{z_t} \\ m_{z_t} \end{bmatrix}, \ \Sigma \Big) \\
&\sim \mathcal{N}\Big(\begin{bmatrix} m_{b_{tt}} \\ m_{z_{tt}} \end{bmatrix}, \ \Sigma \Big) \\
m_{b_{tt}} &= S_{b|y}(S_\epsilon^{-1} y_t + S_{b_t}^{-1} m_{z_t}) \\
m_{z_{tt}} &= m_{z_t} + C_{bz|y}^\top S_\epsilon^{-1}(y_t - m_{z_t}) \\
p(b_t|z_t, y_t) &\sim \mathcal{N}\Big(m_{b_{tt}} + C_{bz|y} S_{z|y}^{-1}(z_t - m_{z_{tt}}), \ S_{b|y} - C_{bz|y} S_{z|y}^{-1} C_{bz|y}^\top \Big) \\
&\sim \mathcal{N}\Big(\underbrace{w_z z_t + w_y y_t}_{Z_{tt}}, \ \underbrace{(S_\epsilon^{-1} + V_t^{-1})^{-1}}_{V_{tt}} \Big)
\end{aligned} \tag{8}$$

The policy input is currently the mean-belief Z_{tt}, a function of Z_t and Y_t. If Z_t or Y_t are random, then Z_{tt} is random too,

$$\begin{aligned}
Z_{tt} &= w_z Z_t + w_y Y_t \\
\mathbb{E}_{zy}[Z_{tt}] &= w_z m_{z_t} + w_y m_{x_t} \\
\mathbb{V}_{zy}[Z_{tt}] &= [w_z, w_y] \begin{bmatrix} S_{z_t} & C_{zx} \\ C_{zx}^\top & S_{x_t} + S_\epsilon \end{bmatrix} [w_z, w_y]^\top
\end{aligned}$$

We assume the policy has linear function (or can be approximated as such):

$$U_t = \theta^\top \begin{bmatrix} Z_{tt} \\ Z_{tt}^\circ \end{bmatrix}$$

and have a policy call:

$$[m_u, S_u, C_u] = \mathtt{policy}(\mathbb{E}_{zy}[Z_{tt}], \mathbb{V}_{zy}[Z_{tt}]) \quad \text{(where } C_u \text{ is } C_{poli}\text{)} \tag{9}$$

The using results from Appendix **??** we have:

$$\begin{aligned}
\mathbb{C}\left[\begin{bmatrix} Z_{tt} \\ Z_{tt}^\circ \end{bmatrix}, U_t \right] = \begin{bmatrix} \mathbb{C}[Z_{tt}, U_t] \\ \mathbb{C}[Z_{tt}^\circ, U_t] \end{bmatrix} &= \begin{bmatrix} S_{Z_{tt}} & \mathbb{C}[Z_{tt}, Z_{tt}^\circ] \\ \mathbb{C}[Z_{tt}^\circ, Z_{tt}] & S_{Z_{tt}^\circ} \end{bmatrix} \theta \\
\therefore \mathbb{C}[Z_{tt}, U_t] &= [S_{Z_{tt}}, \mathbb{C}[Z_{tt}, Z_{tt}^\circ]]\theta \\
&= S_{Z_{tt}} \underbrace{[I, C_{gtrig}[Z_{tt}, Z_{tt}^\circ]]}_{g} \theta \\
\mathbb{C}(X_t, U_t) &= \mathbb{C}(X_t, Z_{tt}) g\theta = [C_{xz}, S_{x_t}] w^\top g\theta \\
\mathbb{C}(Y_t, U_t) &= \mathbb{C}(Y_t, Z_{tt}) g\theta = [C_{xz}, S_{y_t}] w^\top g\theta \\
\mathbb{C}(Z_t, U_t) &= \mathbb{C}(Z_t, Z_{tt}) g\theta = [S_{z_t}, C_{xz}^\top] w^\top g\theta
\end{aligned}$$

Since $\mathbb{C}(Z_{tt}, U_t) = \mathbb{C}(Z_{tt}, Z_{tt})\theta = S_{z_{tt}}\theta$, and $C_u \triangleq S_{z_{tt}}^{-1}\mathbb{C}(Z_{tt}, U_t)$, then we have: $\theta = C_u$.

In summary:
We start with prior joint:

$$p\begin{pmatrix} x_t \\ y_t \\ b_t \\ z_t \end{pmatrix} \quad \sim \quad \left(\begin{bmatrix} m_{x_t} \\ m_{x_t} \\ m_{z_t} \\ m_{z_t} \end{bmatrix}, \begin{bmatrix} S_{x_t} & S_{x_t} & C_{xz} & C_{xz} \\ S_{x_t} & S_{x_t} + S_\epsilon & C_{xz} & C_{xz} \\ C_{xz}^\top & C_{xz}^\top & S_{z_t} + V_t & S_{z_t} \\ C_{xz}^\top & C_{xz}^\top & S_{z_t} & S_{z_t} \end{bmatrix}\right) \tag{10}$$

We then condition on observation y_t, and then decide action u_t:

$$p\begin{pmatrix} x_t|y_t \\ b_t|y_t \\ z_t|y_t \\ u_t|y_t \end{pmatrix} \quad \sim \quad \left(\begin{bmatrix} S_\epsilon S_{y_t}^{-1} m_{x_t} + S_{x_t} S_{y_t}^{-1} y_t \\ m_{b_{tt}} \\ m_{z_{tt}} \\ m_u \end{bmatrix}, \begin{bmatrix} (S_{x_t}^{-1} + S_\epsilon^{-1})^{-1} & S_\epsilon S_{y_t}^{-1} C_{xz} & S_\epsilon S_{y_t}^{-1} C_{xz} & [C_{xz}, S_{x_t}] w^\top g C_u \\ C_{xz}^\top S_{y_t}^{-1} S_\epsilon & S_{b|y} & C_{bz|y} & [S_{z_t}, C_{xz}^\top] w^\top g C_u \\ C_{xz}^\top S_{y_t}^{-1} S_\epsilon & C_{bz|y}^\top & S_{z|y} & [S_{z_t}, C_{xz}^\top] w^\top g C_u \\ C_u^\top g^\top w [C_{xz}, S_{x_t}]^\top & C_u^\top g^\top w [S_{z_t}, C_{xz}^\top]^\top & C_u^\top g^\top w [S_{z_t}, C_{xz}^\top]^\top & S_u \end{bmatrix}\right)$$

Problem: If we instead compute $p(b_t, z_t|y_t)$ by looking only at at Eq. 10 and then conditioning, then we get a different answer:

$$p(b_t, z_t|y_t) \quad \sim \quad \mathcal{N}\left(\begin{bmatrix} m_{z_t} + C_{xz}^\top S_{y_t}^{-1}(y_t - m_{x_t}) \\ m_{z_t} + C_{xz}^\top S_{y_t}^{-1}(y_t - m_{x_t}) \end{bmatrix}, \begin{bmatrix} S_{b_t} - C_{xz}^\top S_{y_t}^{-1} C_{xz} & S_{z_t} - C_{xz}^\top S_{y_t}^{-1} C_{xz} \\ S_{z_t} - C_{xz}^\top S_{y_t}^{-1} C_{xz} & S_{z_t} - C_{xz}^\top S_{y_t}^{-1} C_{xz} \end{bmatrix}\right) \tag{11}$$

where:

$$\begin{aligned} S_{y_t} &= S_{x_t} + S_\epsilon \\ S_{b_t} &= S_{z_t} + V_t \\ S_{b|y} &= (S_\epsilon^{-1} + S_{b_t}^{-1})^{-1} \\ C_{bz|y} &= S_{b|y} S_{b_t}^{-1} S_{z_t} \\ S_{z|y} &= S_{z_t} - S_{z_t}(S_{b_t} + S_\epsilon)^{-1} S_{z_t} \\ w &= [w_z, w_y] = [S_\epsilon(S_\epsilon + V_t)^{-1},\ V_t(S_\epsilon + V_t)^{-1}] \end{aligned}$$

TODO - integrate over y_t..., before or after prediction step?