

Chapter 2

A Bayesian Approach to Haptic Teleoperation Systems

Nomenclatures

t	Continuous time at the master manipulator
t_s	Continuous time at the slave manipulator
k	Discrete time of master clock
k_s	Discrete time of slave clock
k_d	Delayed time of master clock
k_{ds}	Delayed time of slave clock
n_m	Delay variable from the master
n_s	Delay variable from the slave
$\mathbf{x}_m$	Master position
$\mathbf{x}_s$	Position received at the slave
$\hat{\mathbf{x}}_m$	Estimated/predicted master position at the slave
$\mathbf{f}_s$	Slave force
$\mathbf{f}_m$	Force received at the slave
$\hat{\mathbf{f}}_m$	Estimated/predicted slave force at the master
w_m	Importance weight of the master
$\tilde{w}_m$	Normalized importance weight of the master
w_s	Importance weight of the slave
$\tilde{w}_s$	Normalized importance weight of the slave
N	Number of particles
q	Importance distribution
$\mathbf{g}_1, \mathbf{g}_2$	State transition functions
$\mathbf{h}_1, \mathbf{h}_2$	Observation transition functions
$\mathbf{l}_1, \mathbf{l}_2$	State noises
$\mathbf{r}_1, \mathbf{r}_2$	Observation noises

Unreliable network behaviors, such as time-varying delay, packet loss, and other path characteristics, are in general described by random processes. In the real network environment, delay jitter is observed since packets are transmitted through multiple intermediate nodes with random queuing delays. The delay increases randomly when packets are congested over the network with limited bandwidth.

Consequently, such congested network causes packet loss, which is also a random behavior. In this chapter, based on the stochastic nature of the network, we present a stochastic representation for haptic data, which may be time-varying delayed or lost over the network. A Bayesian approach,¹ which models uncertainty as stochastic variables, is proposed to estimate or predict haptic position and force data through unreliable network conditions. We also propose a practical solution of the Bayesian approach, which is known as a robust tracking method for nonlinear problems.

This chapter is organized as follows. In Sect. 2.1, we present a state-space representation of haptic data over the unreliable network in order to perform the stochastic estimation or prediction. Based on the state-space representation, a Bayesian approach that finds the optimal estimates of position and force data is presented in Sect. 2.2. As a practical solution, the Bayesian particle filter is proposed in Sect. 2.3 in order to estimate or predict position and force data over unreliable network conditions.

2.1 State-Space Representations of Haptic Data Over the Network

A haptic teleoperation system or a haptic interface that exchanges position and force data over the network is shown in Fig. 2.1. In the *uplink transmission* of the network, position data from a master manipulator are transmitted to a slave manipulator through the network. Based on the movements of the master manipulator, reflected force data are generated by any contact of the slave manipulator with the remote environment, and then they are fed into the master manipulator in the *downlink transmission*. The master and slave manipulators, which are interfaced with the human operator and remote environment, respectively, are continuous systems, whereas position and force data transmitted through the network are sampled data in discrete time. When the discrete position and force data are transmitted over unreliable network conditions, such as time-varying delay and packet loss, they can be formulated in nonlinear state-space models to perform the proposed stochastic estimation or prediction.

2.1.1 Uplink Transmission

In the uplink transmission of Fig. 2.1, $\mathbf{x}_m[k]$, which is sampled from a continuous movement $\mathbf{x}_h[t]$, is multiple-degree of freedom (m-DoF) true position vector in a discrete time k , where k represents a time instance of local clock at the master manipulator. The master position $\mathbf{x}_m[k]$ is transmitted through the network, and the

¹Basics of the Bayesian statistics are presented in Appendix A.

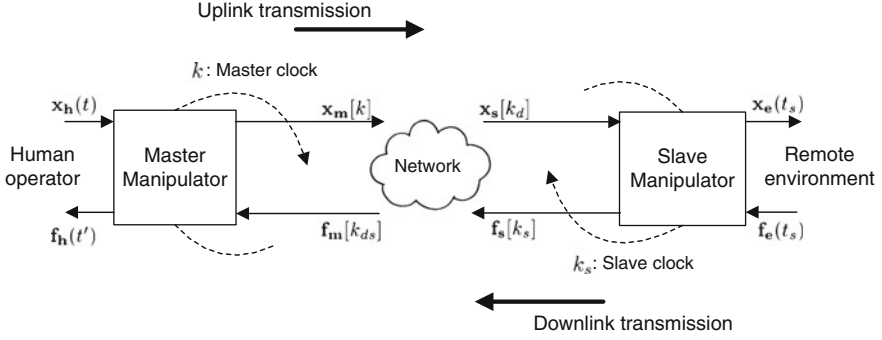


Fig. 2.1 A haptic teleoperation system that exchanges position and force data over the network

received position at the slave manipulator is represented as $\mathbf{x}_s[k_d]$, where k_d is a delayed time instance from the master manipulator. Since the network delay varies over time, k_d is given by

$$k_d = k + n_m[k] \quad (2.1)$$

where $n_m[k]$ represents a time-varying delay in the uplink transmission, and it can be expressed as a function of time k . Note that when the network delay is constant, $n_m[k]$ is not necessarily a function of time k , so (2.1) becomes $k_d = k + n_m$. Hence, the delayed position at the slave manipulator is $\mathbf{x}_s[k + n_m] = \mathbf{x}_m[k]$, which can be a deterministic signal rather than a stochastic variable. In order to perform the stochastic estimation or prediction, we first define a state-space model of position data flow through the network. The state vector is given by

$$\mathbf{x}_m[k] = \mathbf{g}_1(\mathbf{x}_m[k-1] + \mathbf{l}_1[k-1]) \quad (2.2)$$

where $\mathbf{x}_m[k]$ and $\mathbf{x}_m[k-1]$ are the current and previous states, respectively. $\mathbf{l}_1[k-1]$ is the state noise, which may be a nonnormal² distribution. $\mathbf{g}_1(\cdot)$ is the nonlinear state transition function, which represents the human movement from the master manipulator. In the stochastic estimation or prediction problem, the objective is to find the state using *observations*.³ The observation vector is given by

$$\mathbf{x}_s[k_d] = \mathbf{h}_1(\mathbf{x}_m[k] + \mathbf{r}_1[k_d]) \quad (2.3)$$

where $\mathbf{x}_s[k_d]$ is the observation vector, $\mathbf{r}_1[k_d]$ is the observation noise, and $\mathbf{h}_1(\cdot)$ is the nonlinear observation function. The state and observation processes in the uplink transmission are visualized in Fig. 2.2. In this figure, the master positions $\{\mathbf{x}_m[k-1], \mathbf{x}_m[k], \mathbf{x}_m[k+1]\}$ are the hidden states, which are not directly

²A random variable with a Gaussian distribution is said to be normally distributed.

³An observation is also called a measurement.

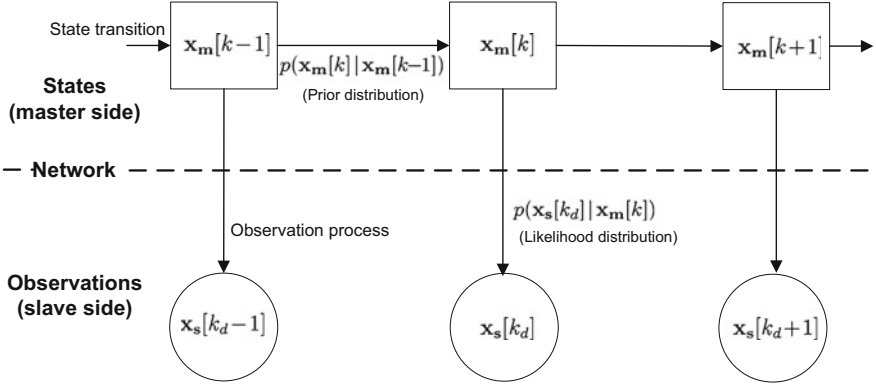


Fig. 2.2 State and observation processes of position data in the uplink transmission. The current state at the master manipulator depends on the previous state, and the current observation depends on the current state

observable from the slave manipulator. The current state depends only on the previous state⁴ so that a state transition in (2.2) is described by a *prior* distribution, which is $p(\mathbf{x}_m[k]|\mathbf{x}_m[k-1])$. The received positions $\{\mathbf{x}_s[k_d-1], \mathbf{x}_s[k_d], \mathbf{x}_s[k_d+1]\}$ are the observations at the slave manipulator, which depend on the states. This observation process in (2.3) is described by a *likelihood* distribution, which is $p(\mathbf{x}_s[k_d]|\mathbf{x}_m[k])$.⁵ Note that since this position data flow is modeled based on a Markov process with unobserved hidden states, such stochastic model is also called a *hidden Markov model* [2].

When the delayed position data are received at the slave manipulator, they need to be represented by time instances of local clock at the slave manipulator, which are $\{k_s-1, k_s, k_s+1\}$. Assuming that the clocks of master and slave manipulators are synchronized, the representations of delayed position data at the slave manipulator are shown in Fig. 2.3. In the presence of constant delay, which is shown in Fig. 2.3a, we simply have $k_s = k_d$ because k_d is not actually varying. Note that k_{d_0} is the delay of initial master position, which is given by $k_{d_0}k_{d_0} = n_m[0]$. When the time-varying delay is introduced, which is shown in Fig. 2.3b, the sample $\mathbf{x}_m[k+1]$ is further delayed in this example. While the slave clock runs at its sampling rate (e.g. 1 kHz), the slave manipulator holds the last received sample $\mathbf{x}_s[k_d]$ so that $\mathbf{x}_s[k_s+1] = \mathbf{x}_s[k_d]$ in this case. Therefore, in the presence of time-varying delay, we can have $k_s \neq k_d$. For the last case, the packet loss behavior is illustrated in Fig. 2.3c. In this example, the sample $\mathbf{x}_s[k_d]$ is lost during the network transmission.

⁴This is a first-order Markov process. The current state does not depend on previous trajectory, but only the previous state. i.e. $p(\mathbf{x}_m[k]|\mathbf{x}_m[k-1], \mathbf{x}_m[k-2], \dots, \mathbf{x}_m[0]) = p(\mathbf{x}_m[k]|\mathbf{x}_m[k-1])$.

⁵ $p(\mathbf{x}_m[k]|\mathbf{x}_m[k-1])$ describes the distribution of the current state conditioned on the previous state whereas $p(\mathbf{x}_s[k_d]|\mathbf{x}_m[k])$ describes the distribution of the current observation conditioned on the current state.

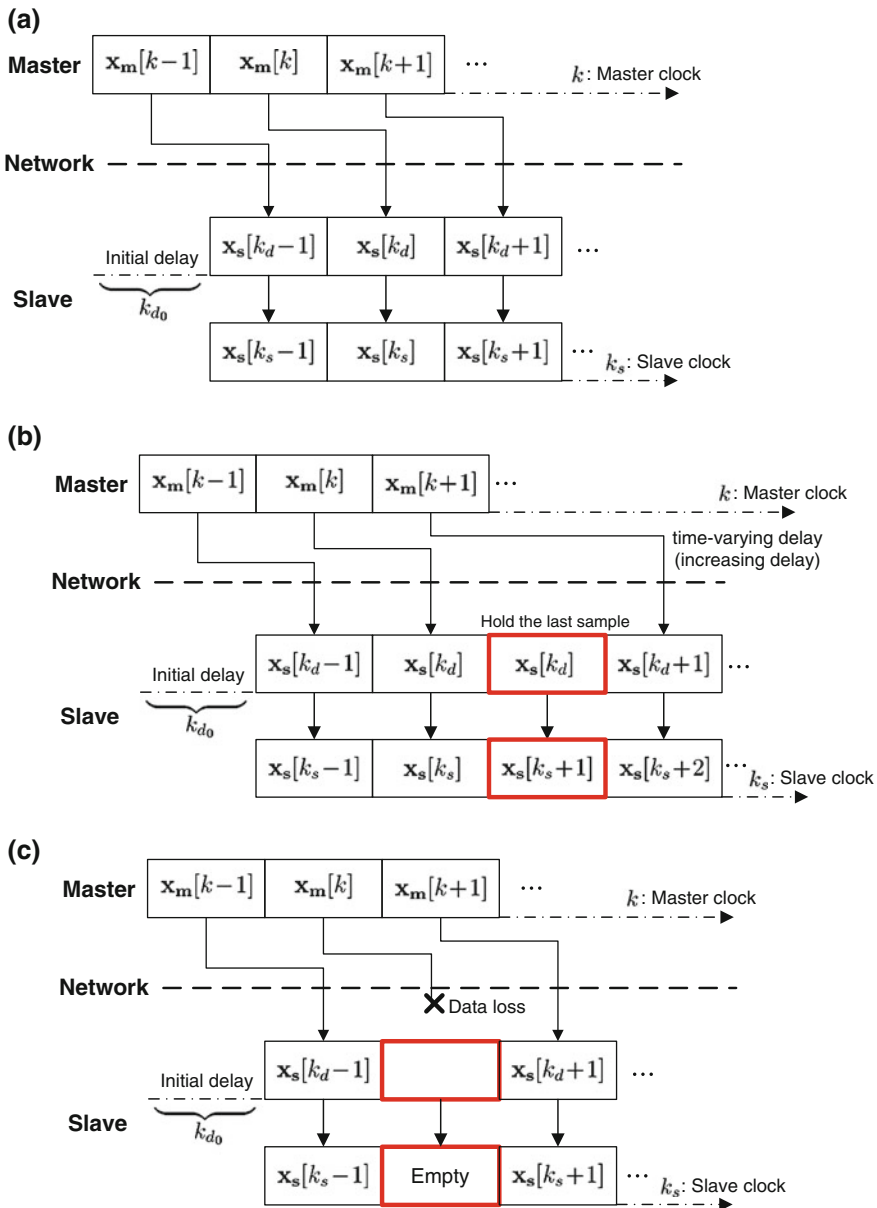


Fig. 2.3 Representations of delayed position data at the slave manipulator. **a** Constant delay case. **b** Time-varying delay (increasing delay) case. In this example, the slave manipulator holds the last received sample. **c** Packet loss case. In this example, the slave manipulator does not hold the last received sample, so that $x_s[k_s] = 0$

While the slave clock runs at its sampling rate, the slave manipulator uses the strategy that the unreceived sample becomes null data, which is $\mathbf{x}_s[k_s] = 0$. Note that in this packet loss case, it is also possible to choose the strategy used in Fig. 2.3b so that $\mathbf{x}_s[k_s] = \mathbf{x}_s[k_d - 1]$, instead of $\mathbf{x}_s[k_s] = 0$.

2.1.2 Downlink Transmission

A force data flow in the downlink transmission of the network is shown in Fig. 2.1. Once a slave position is obtained at the discrete time instance k_s , the slave manipulator interacts with the remote environment by converting to a continuous position $\mathbf{x}_e[t_s]$. When position and force data are transmitted with an equal sampling rate (e.g. 1 kHz), a continuous force $\mathbf{f}_e[t_s]$ is generated by any contact with the remote environment and is sampled as $\mathbf{f}_s[k_s]$ for transmitting over the network. The received force at the master manipulator, which is delayed over the network, is $\mathbf{f}_m[k_{ds}]$, and the delayed time instance $[k_{ds}]$ is given by

$$k_{ds} = k_s + n_s[k_s]. \quad (2.4)$$

$n_s[k_s]$ represents a time-varying delay in the downlink transmission, which is expressed as a function of time k_s . When $n_s[k_s]$ is a constant delay, $n_s[k_s]$ is not varying over time so that (2.4) becomes $k_{ds} = k_s + n_s$. Hence, the delayed force at the master manipulator is $\mathbf{f}_m[k_s + n_s] = \mathbf{f}_s[k_s]$, which can be a deterministic signal rather than a stochastic variable. In order to perform the proposed stochastic estimation or prediction, we first define a state-space model of force data flow in the downlink transmission. The state vector is given by

$$\mathbf{f}_s[k_s] = \mathbf{g}_2(\mathbf{f}_s[k_s - 1] + \mathbf{l}_2[k_s - 1]) \quad (2.5)$$

where $\mathbf{f}_s[k_s]$ and $\mathbf{f}_s[k_s - 1]$ are the current and previous states, respectively. $\mathbf{l}_2[k_s - 1]$ is the state noise that may be a nonnormal distribution. $\mathbf{g}_2(\cdot)$ is the nonlinear state transition function, which represents the force trajectory from the slave manipulator. The observation vector is given by

$$\mathbf{f}_m[k_{ds}] = \mathbf{h}_2(\mathbf{f}_s[k_s] + \mathbf{r}_2[k_{ds}]) \quad (2.6)$$

where $\mathbf{f}_m[k_{ds}]$ is the observation vector, $\mathbf{r}_2[k_{ds}]$ is the observation noise, and $\mathbf{h}_2(\cdot)$ is the nonlinear observation function. As illustrated for the state and observation processes of position data in Fig. 2.2, the slave force $\mathbf{f}_s[k_s]$ is the hidden state and is not directly observable from the master manipulator. The current state only depends on the previous state, which is described by $p(\mathbf{f}_s[k_s]|\mathbf{f}_s[k_s - 1])$. The observation vector $\mathbf{f}_m[k_{ds}]$, which is the received force at the master manipulator, depends on the state, described by $p(\mathbf{f}_m[k_{ds}]|\mathbf{f}_s[k_s])$.

In order to represent the delayed force in terms of the time instance of master clock, we use the strategies shown in Fig. 2.3. In the presence of constant delay, because k_{ds} is not varying, we simply have $k' = k_{ds}$. Note that k' is a delayed time instance of the master clock, which may be expressed as $k' = k + k_{ds0}$, and k_{ds0} is the delay of initial slave force such as $k_{ds0} = n_s[0]$. In the presence of time-varying delay, we can have $k' \neq k_{ds}$ as shown in the example of Fig. 2.3b. When the packet loss occurs, as presented in Fig. 2.3c, the unreceived force sample becomes null data, which becomes zero.

2.2 A Bayesian Approach for Haptic Teleoperation Systems

Given the state-space models for the uplink transmission (2.2) and (2.3), and the downlink transmission (2.5) and (2.6), the objective is to find the states by using the stochastic estimation or prediction. In this section, a Bayesian approach is considered, in which the estimation or prediction is recursively performed by computing the states given available observations. A haptic teleoperation system based on the estimation or prediction method is shown in Fig. 2.4. In this figure, the estimates of the states in the uplink and downlink transmissions are denoted as $\hat{\mathbf{x}}_m$ and $\hat{\mathbf{f}}_s$, respectively.

2.2.1 Uplink Transmission

In a Bayesian approach, the estimate of position data $\hat{\mathbf{x}}_m[k_s]$ can be obtained using the previous state $\mathbf{x}_m[k_s - 1]$ and available observation $\mathbf{x}_s[k_{d0}:k_d]$. Note that $\mathbf{x}_s[k_{d0}:k_d]$ represents the received positions at the slave manipulator from the initial time k_{d0} to k_d . The process to find the state $\hat{\mathbf{x}}_m[k_s]$ based on the state-space model of the uplink transmission is shown in Fig. 2.5.

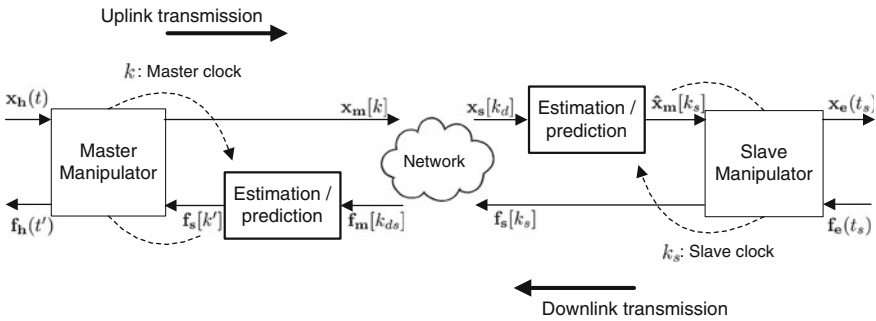


Fig. 2.4 A haptic teleoperation system based on the prediction or estimation to compensate for unreliable network conditions, such as time-varying delay or packet loss

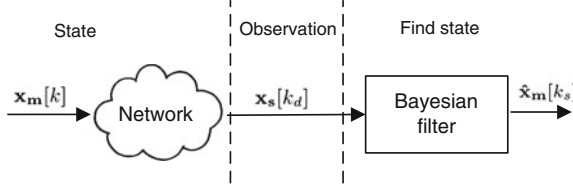


Fig. 2.5 A Bayesian approach to find the state, given the observation in the uplink transmission

In this Bayesian approach, since the state is computed given the observations, it is a problem to construct a *posterior* distribution, which is $p(\mathbf{x}_m[k_s]|\mathbf{x}_s[k_{d_0}:k_d])$. A posterior distribution is a distribution of an unknown quantity, which is conditioned on an observation. The optimal solution that finds the state using the posterior distribution is given by [3, 7]

$$\mathbb{E}(\mathbf{x}_m[k_s]|\mathbf{x}_s[k_{d_0}:k_d]) = \int \mathbf{x}_m[k_s] p(\mathbf{x}_m[k_s]|\mathbf{x}_s[k_{d_0}:k_d]) d\mathbf{x}_m \quad (2.7)$$

where $\mathbb{E}(\cdot)$ denotes the expectation. The posterior distribution in (2.7) is recursively computed by using the following two steps [3, 7]:

$$p(\mathbf{x}_m[k_s]|\mathbf{x}_s[k_{d_0}:k_d - 1]) = \int p(\mathbf{x}_m[k_s]|\mathbf{x}_m[k_s - 1]) p(\mathbf{x}_m[k_s - 1]|\mathbf{x}_s[k_{d_0}:k_d - 1]) d\mathbf{x}_m \quad (2.8)$$

$$p(\mathbf{x}_m[k_s]|\mathbf{x}_s[k_{d_0}:k_d]) = \frac{p(\mathbf{x}_s[k_d]|\mathbf{x}_m[k_s]) p(\mathbf{x}_m[k_s]|\mathbf{x}_s[k_{d_0}:k_d - 1])}{p(\mathbf{x}_s[k_d]|\mathbf{x}_s[k_{d_0}:k_d - 1])} \quad (2.9)$$

Equations (2.8) and (2.9) are called the *prediction* and *update* steps, respectively. The prediction step (2.8) is evolved from the posterior distribution at the previous time instance, which is $p(\mathbf{x}_m[k_s - 1]|\mathbf{x}_s[k_{d_0}:k_d - 1])$. Then, the update step (2.9) is evolved from the prediction step (2.8), given the observation at the current time instance k_d .

In general, the computations of (2.8) and (2.9) are not directly solvable. Hence, Bayesian filters, such as the Kalman filter and the Bayesian particle filter, are used in real-world applications for the practical computation of the posterior distribution. In the case of Kalman filter, which sequentially deduces the optimal solution, we should assume that the state noise $\mathbf{I}_1[k]$ in (2.2) and observation noise $\mathbf{r}_1[k_d]$ in (2.3) have normal distributions. Furthermore, the state and observation transition functions $\mathbf{g}_1(\cdot)$ and $\mathbf{h}_1(\cdot)$ need to be assumed as linear equations. Given these assumptions, the estimates of position and force data can be obtained by sequentially deducing the distributions in (2.8) and (2.9) of the optimal solution [4].

When the transition functions are nonlinear or the noise sequences are not normally distributed, which may be observed more frequently in real-world

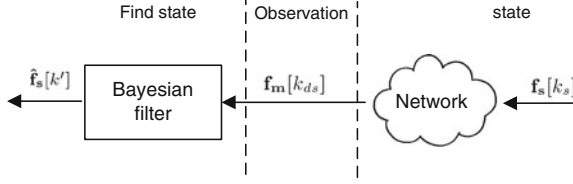


Fig. 2.6 A Bayesian approach to find the state, given the observation in the downlink transmission

applications, the Bayesian particle filter provides a robust tracking performance [5]. In this thesis, we focus our attention on the Bayesian particle filter to estimate or predict position and force data. A haptic teleoperation system using the proposed Bayesian particle filter is presented in Sect. 2.3.

2.2.2 Downlink Transmission

In the downlink transmission, the estimate of force data $\hat{\mathbf{f}}_s[k']$ can be obtained using the previous state $\mathbf{f}_s[k' - 1]$ and available observation $\mathbf{f}_m[k_{ds_0} : k_{ds}]$, where $\mathbf{f}_m[k_{ds_0} : k_{ds}]$ represents the received forces at the master manipulator from the initial time k_{ds_0} to k_{ds} . The process to find the state $\hat{\mathbf{f}}_s[k']$ based on the state-space model of the downlink transmission is shown in Fig. 2.6.

Similar to the uplink transmission, the optimal solution of force data based on the Bayesian approach is given by

$$\mathbb{E}(\mathbf{f}_s[k'] | \mathbf{f}_m[k_{ds_0} : k_{ds}]) = \int \mathbf{f}_s[k'] p(\mathbf{f}_s[k'] | \mathbf{f}_m[k_{ds_0} : k_{ds}]) d\mathbf{f}_s. \quad (2.10)$$

This optimal solution (2.10) can be obtained by recursively computing the prediction step $p(\mathbf{f}_s[k'] | \mathbf{f}_m[k_{ds_0} : k_{ds} - 1])$ and the update step $p(\mathbf{f}_s[k'] | \mathbf{f}_m[k_{ds_0} : k_{ds}])$, as presented in (2.8) and (2.9). Since we assume that the transition functions are nonlinear or noises are not necessarily normal distributions, such as (2.5) and (2.6), we focus our attention on the Bayesian particle filter for tracking force data in the downlink transmission.

2.3 Bayesian Particle Filter for Haptic Teleoperation Systems

In the Bayesian particle filter, the required posterior distributions in (2.7) and (2.10) are represented by a set of random samples with associated weights. This is a sequential Monte Carlo method that requires a number of random samples in order

to obtain the estimated or predicted states. As a number of samples increases, this Monte Carlo characteristic represents the required posterior distribution, and therefore, the Bayesian particle filter approaches to the optimal solution.

2.3.1 Uplink Transmission

Using the Bayesian particle filter, the posterior distribution in the uplink transmission (2.9) is empirically approximated by drawing a set of randomly chosen weighted samples such that [5]

$$\begin{aligned} p(\mathbf{x}_m[k_s]|\mathbf{x}_s[k_{d_0}:k_d]) &\approx \frac{1}{N} \sum_{i=1}^N \tilde{w}_m^i[k_s] \delta(\mathbf{x}_m[k_{s_0}:k_s] - \mathbf{x}_m^i[k_{s_0}:k_s]) \\ &\triangleq \hat{p}(\mathbf{x}_m[k_s]|\mathbf{x}_s[k_{d_0}:k_d]) \end{aligned} \quad (2.11)$$

where $\delta(\cdot)$ is the Dirac delta function, N is the number of particles, and $\hat{p}(\cdot)$ is the approximation of posterior distribution. A particle refers a sample in order to represent a set of states according to probability distributions. In (2.11), the posterior distribution can be represented by the set of random samples $\{\mathbf{x}_m^i[k_{s_0}:k_s]\}_{i=1}^N$ with associated weights $\{w_m^i[k_s]\}_{i=1}^N$, where time instance k_{s_0} denotes the initial time instance of slave local clock. The weight $w_m^i[k_s]$ is called the *importance weight*, which is given by

$$w_m^i[k_s] = \frac{p(\mathbf{x}_m^i[k_{s_0}:k_s]|\mathbf{x}_s[k_{d_0}:k_d])}{q(\mathbf{x}_m^i[k_{s_0}:k_s]|\mathbf{x}_s[k_{d_0}:k_d])} \quad (2.12)$$

where $q(\cdot)$ in the denominator is called an importance distribution, and $\tilde{w}_m^i[k_s]$ in (2.11) is a normalized importance weight given by

$$\tilde{w}_m^i[k_s] = w_m^i[k_s] / \sum_{i=1}^N w_m^i[k_s]. \quad (2.13)$$

Computing the importance weight (2.12) involves the whole previous trajectory of state, which may require high computational complexity. Hence, we reformulate (2.12) in a recursive form by factorizing⁶ $q(\cdot)$ such that

$$\begin{aligned} q(\mathbf{x}_m[k_{s_0}:k_s]|\mathbf{x}_s[k_{d_0}:k_d]) &= q(\mathbf{x}_m[k_s]|\mathbf{x}_m[k_s-1], \mathbf{x}_s[k_{d_0}:k_d]) \\ &\quad \cdot q(\mathbf{x}_m[k_{s_0}:k_s-1]|\mathbf{x}_s[k_{d_0}:k_d-1]) \end{aligned} \quad (2.14)$$

⁶Factorization of joint distribution is given by $p(a,b|c) = p(a|b,c)p(b|c)$.

The numerator in (2.12) can be also written in a factorized form that allows computing the posterior distribution sequentially. Based on these factorized forms, a new recursive and time-update form of the importance weight can be computed as

$$w_m^i[k_s] = \tilde{w}_m^i[k_s - 1] \frac{p(\mathbf{x}_s[k_d] | \mathbf{x}_m^i[k_s]) p(\mathbf{x}_m^i[k_s] | \mathbf{x}_m^i[k_s - 1])}{q(\mathbf{x}_m^i[k_s] | \mathbf{x}_m^i[k_s - 1], \mathbf{x}_s[k_d])}. \quad (2.15)$$

This importance weight is associated with the state $\mathbf{x}_m^i[k_s]$ at time k_s for i th particle. After the factorization, the previous trajectory of state is not necessarily computed. Hence, the posterior distribution in (2.11) can be simplified as

$$p(\mathbf{x}_m[k_s] | \mathbf{x}_s[k_{d_0}:k_d]) \approx \frac{1}{N} \sum_{i=1}^N \tilde{w}_m^i[k_s] \delta(\mathbf{x}_m[k_s] - \mathbf{x}_m^i[k_s]) \quad (2.16)$$

where the importance weight is (2.15). The recursive computation (2.15) depends on the choice of the importance distribution $q(\cdot)$, which is the denominator of (2.15). One of the most popular choices is to employ a prior distribution such as

$$q(\mathbf{x}_m^i[k_s] | \mathbf{x}_m^i[k_s - 1], \mathbf{x}_s[k_d]) = p(\mathbf{x}_m^i[k_s] | \mathbf{x}_m^i[k_s - 1]). \quad (2.17)$$

Substituting (2.17) into (2.15), Eq. (2.15) is simplified as

$$w_m^i[k_s] = \tilde{w}_m^i[k_s - 1] p(\mathbf{x}_s[k_d] | \mathbf{x}_m^i[k_s]). \quad (2.18)$$

Using (2.18), implementing the proposed Bayesian particle filter can be more easily achieved, since only the likelihood distribution in (2.18) needs to be computed in order to compute the approximated posterior distribution (2.16).

One of the issues using the Bayesian particle filter in practice is that the recursive computation (2.16) may diverge as iterations proceed. In order to avoid such divergence problem, the *resampling* step needs to be performed by using the associated importance weights. When cumulative distribution functions (CDFs) of the normalized weights $\{\tilde{w}_m^i[k_s]\}_{i=1}^N$ are constructed, each element of the CDF is compared with a uniformly distributed function to determine whether the weights are relatively high or low. Based on this determination, the samples with low weights are eliminated, and the samples with high weights are retained for estimating or predicting the states. Therefore, a new set of states $\{\mathbf{x}_m^i[k_s]\}_{i=1}^N$ is generated, and this procedure is repeated for the next time instance. A general process of the Bayesian particle filter including the resampling step is also presented in Appendix A.3.

2.3.2 Downlink Transmission

Using the Bayesian particle filter, the posterior distribution of the downlink transmission in (2.10) is also empirically approximated by drawing a set of randomly chosen weighted samples such that

$$\hat{p}(\mathbf{f}_s[k'] | \mathbf{f}_m[k_{ds_0}:k_{ds}]) = \frac{1}{N} \sum_{i=1}^N \tilde{w}_s^i[k'] \delta(\mathbf{f}_s[k'] - \mathbf{f}_s^i[k']). \quad (2.19)$$

The importance weight in (2.19) can be computed recursively by selecting the importance distribution $q(\cdot)$ as a prior distribution. Hence, the simplified form of the importance weight for the downlink transmission is given by

$$w_s^i[k'] = \tilde{w}_s^i[k' - 1] p(\mathbf{f}_m[k_{ds}] | \mathbf{f}_s^i[k']). \quad (2.20)$$

The implementation to obtain the estimate of force data can be relatively easy in practice, since we only need to compute the likelihood distribution in (2.20). After normalizing the importance weight as $\tilde{w}_s^i[k'] = w_s^i[k'] / \sum_{i=1}^N w_s^i[k']$, the resampling step needs to be performed to regenerate a new set of states $\{\mathbf{f}_s^i[k']\}_{i=1}^N$ based on the associated importance weights. In addition to the uplink transmission, the entire processes of the Bayesian particle filter, including the initialization, prediction, update, resampling, and iteration steps, are described in Table 2.1.

Table 2.1 Bayesian particle filters for the uplink and downlink transmissions

<i>1. Initialization step</i>
Initialize states $\{\mathbf{x}_m^i[k_{s_0}]\}_{i=1}^N$, $\{\mathbf{f}_s^i[k'_0]\}_{i=1}^N$ by initial distributions, which may be known or chosen randomly
<i>2. Prediction step</i>
Predict states $\mathbf{x}_m[k_s + 1]$, $\mathbf{f}_s[k' + 1]$, by distributions $p(\mathbf{x}_m[k_s + 1] \mathbf{x}_m^i[k_s])$, $p(\mathbf{f}_s[k' + 1] \mathbf{f}_s^i[k'])$
<i>3. Update step</i>
Compute importance weights $\{\tilde{w}_m^i[k_s]\}_{i=1}^N$ (2.18) and $\{\tilde{w}_s^i[k']\}_{i=1}^N$ (2.20)
<i>4. Resampling step</i>
Draw N new states according to associated importance weights as $\{\mathbf{x}_m^i[k_s]\}_{i=1}^N$ and $\{\mathbf{f}_s^i[k']\}_{i=1}^N$
<i>5. Iteration step</i>
Set $k = k + 1$, $k_s = k_s + 1$ and repeat from Step (2)

2.4 Discussion

The Bayesian particle filter introduced in Sect. 2.3 can be used in either prediction or estimation depending on the availability of observation. When a sample is lost during the network transmission, the observation at the corresponding time instance is not available at the receiving manipulator. In this case, the proposed Bayesian particle filter performs the prediction using the observations up to the past time instances. This practical approach of the Bayesian particle filter in the case of the packet loss and time-varying delay is presented in Chap. 4.

In Sect. 2.3, the Bayesian particle filter is applied only to find the estimates of position and force data. In the current network technology based on the Internet, the delay, which is time varying depending on network traffic conditions, is often expected to be estimated or predicted for further improvement in performance [1, 6]. Using the Bayesian particle filter, the estimate of network delay in addition to position and force data is considered and presented in Appendix B. In the uplink transmission, a concatenated vector, which consists of the position vector and the time-varying delay $n_m[k]$, is constructed and processed by the Bayesian particle filter in order to solve a multivariable estimation problem. A similar process is performed for the downlink transmission by constructing a concatenated vector of the time-varying delay $n_s[k_s]$ and force data.

References

1. T. Mirfakhrai, S. Payandeh, On using delay prediction in controlling force reflecting teleoperation over the Internet. *Robotica* **23**(6), 809–813 (2005)
2. C.M. Bishop, *Pattern Recognition and Machine Learning* (Springer, New York, 2007)
3. Z. Chen, Bayesian filtering: From Kalman filters to particle filters, and beyond. Adaptive Systems Lab., McMaster Univ., Hamilton, ON, Canada [Online]. Available: http://soma.crl.mcmaster.ca/zhechen/download/ieee_bayesian.ps
4. B.D.O. Anderson, J.B. Moore, *Optimal Filtering* (Prentice-Hall, Englewood Cliffs, 1979)
5. A. Doucet, N. de Freitas, N. Gordon, *Sequential Monte Carlo Methods in Practice* (Springer, New York, 2001)
6. M.J. Coates, R.D. Nowak, Sequential Monte Carlo inference of internal delays in nonstationary communication networks. *IEEE Trans. Signal Process.* **50**, 366–376 (2002)
7. X.-L. Hu, T.B. Schon, L. Ljung, A basic convergence result for particle filtering. *IEEE Trans. Signal Process.* **56**(4), 1337–1348 (2008)

Haptic Teleoperation Systems

Signal Processing Perspective

Lee, J.-y.; Payandeh, S.

2015, XII, 130 p. 69 illus., 26 illus. in color., Hardcover

ISBN: 978-3-319-19556-8