Received: 30 January 2017 | Revised: 12 June 2017 | Accepted: 5 July 2017
DOI: 10.1002/rnc.3896

WILEY

Predictive control for networked systems affected by correlated packet loss

Edwin G.W. Peters[1] | Damian Marelli[2] | Daniel E. Quevedo[3] | Minyue Fu[1]

[1]School of Electrical Engineering and Computer Science, University of Newcastle, New South Wales, Australia

[2]French Argentine International Center for Information and Systems Sciences, National Scientific and Technical Research Council, Rosario, Argentina

[3]Department of Electrical Engineering (EIM-E), Paderborn University, Paderborn, Germany

Correspondence
Edwin G.W. Peters, School of Electrical Engineering and Computer Science, University of Newcastle, New South Wales 2308, Australia.
Email: edwin.g.w.peters@gmail.com

Summary

In this paper, we consider the predictive control problem of designing receding horizon controllers for networked linear systems subject to random packet loss in the controller to actuator link. The packet dropouts are temporarily correlated in the sense that they obey a Markovian transition model. Our design task is to solve the optimal controller that minimizes a given receding horizon cost function, using the available packet loss history. Due to the correlated nature of the packet loss, standard linear quadratic regulator methods do not apply. We first present the optimal control law by considering the correlations. This controller turns out to depend on the packet loss history and would typically require a large lookup table for implementation when the Markovian order is high. To address this issue, we present and compare several suboptimal design approaches to reduce the number of control laws.

KEYWORDS

jump linear systems, networked control, optimal control, packet dropouts, predictive control, stochastic systems

1 | INTRODUCTION

In this work, we consider predictive control to design controllers for dynamic systems where the communication between the controller and the actuators is affected by correlated packet dropouts. Traditionally, the packet dropouts are modeled as an independent and identically distributed (*i.i.d*) random process.[1-7] This model is also extended to include transmission delays.[8-11] For many network setups, the network effects can however not be captured by an *i.i.d* model, since the packet dropouts in the networks often are correlated.[12] A simple method to address correlated packet dropouts is by using a 2-state Markov model, also called the Gilbert-Elliot or *pq* model, which captures packet losses that occur in bursts.[13,14] Here, the probability for the current packet transmission to be successful depends on the transmission outcome of the previously transmitted packet. This model is frequently used for controller synthesis and analysis.[11,15-18]

While the 2-state Markov chain can capture network effects that the *i.i.d* models could not, more accurate network models can be obtained by considering a longer history of packet dropouts.[12,19-21] This results in higher-order Markov models that can capture faster changing network effects more accurately. Konrad et al[12] proposed a network model, where the network is divided in 2 states: 1 "good" state, where no dropouts occur and a "bad" state, where packet dropouts occur. The proposed model models the length of the good and bad state using an exponential distribution. In the example in the work of Konrad et al,[12] the bad state of the network is modeled using a fourth-order Markov chain. Note that the good and bad states in the work of Konrad et al[12] differ from the Gilbert-Elliot models with a good and bad state, as is used in, for example, the work of Peters et al.[15]

Thus, to summarize, a higher-order network model tends to be a more realistic representation of the real networks. We therefore take the view that a controller design that takes the high-order network model into account will result in better performance on real networks. We will therefore present a controller design that can directly utilize these higher-order models. In fact, our controller design method will provide the optimal controller design for any network that can be modeled accurately by the use of conditional probabilities. The optimal predictive controller is designed to minimize a linear quadratic (LQ) cost function that penalizes future states. Here, the future system states are predicted using the model of the system and the network, the current state, and the history of packet arrivals. This results in a control law that is a function of the state and the packet arrival history. A key issue is that, at the time when a control input is computed, the controller does not know whether this input will actually reach the actuator. Such a design problem can fit into the Markov jump linear system framework by assuming that the packet transmission at the current time will be successful.[17,22-25] The difference in our approach is that we take advantage of the structure of the Markov transition matrix, which results in simpler expressions. Designing the control law in this manner leads to controllers that depend on the system state and on the history of packet dropouts. This results in a controller for every possible observation of packet dropouts, which results in high complexity control laws (meaning: many controllers). The complexity increases exponentially for longer packet dropout histories. It also requires significant amounts of memory to be stored in a lookup table in, for example, microcontrollers. At this point, one might ask whether it is necessary to take the entire packet arrival history into account when designing the control law. To tackle this question, we investigate suboptimal reduced complexity control laws that feature fewer controllers and illustrate the loss of performance compared with a full complexity control law through simulation studies. While there exists an extensive literature on performance and complexity trade-offs in terms of minimizing a cost function for estimation over networks with correlated packet dropouts,[26,27] the controller design has rarely been considered in this setting. One main difference between the estimator and the controller design is that the network is part of a closed-loop system. This makes the controller design more difficult.

In this work, we will present a novel optimal controller design that can be applied to any network, where the random packet dropouts can be modeled using conditional distributions. As described above, conditional distributions can capture network effects that more simple 2-state Markov models cannot.[12] This controller is derived by minimizing a stochastic LQ cost function. The packet dropout data are fully utilized in the optimal control design. For high-order packet dropout models, the optimal controller requires a large lookup table to store the control gains, which may be impractical for implementations. This optimal controller is therefore only suitable when the order of the packet dropout model is relatively low. Our second contribution is to present 3 suboptimal methods that trade off the performance of the controller against the controller complexity. The difference between these 3 methods lies in where in the design process the suboptimality is introduced. The performance of the different methods is analyzed using simulation studies where we implement the controllers as model predictive control.

In this work, we perform controller design for systems that feature network uncertainties in the link between the controller and the actuator. This results in control laws that are robust for network conditions, which result in packet dropouts. When the controller design problem further involves system and/or modeling uncertainties, the presented approach can be combined with methods such as an online system identification method or a robust design. An online system identification method, such as the method presented in the work of Shi and Fang,[28] allows for online updating the system model that is used in the controller design. Methods for robust design aim at optimizing the performance for the worst-case system and the network models, see, for example, the works of Wang et al[29] and Quevedo and Nešić.[30,31]

This paper is structured as follows. We will present the system and network in Section 2. The optimal controller design is performed in Section 3. In Section 4, we present methods on how to reduce the controller complexity and introduce 3 controller design methods. Simulation studies are included in Section 5, where the performance of the different controllers is compared. Section 6 draws conclusions.

Notation. We denote the probability and expectation of the random variable X knowing Y by $\mathbf{Pr}\{X|Y\}$ and $\mathbf{E}\{X|Y\}$, respectively. Denote I_N the $N\times N$ identity matrix, $\mathbf{0}_{n\times m}$ a $n\times m$ matrix of zeros, and $diag\{X\}$ the matrix with X on its (block-)diagonal. The indicator function $\mathbb{1}_x(y) = 1$ if $x = y$ when y is a variable, or $x \in y$ when y is a set. Let card (X) denote the number of elements in the set or sequence X. Denote $\|x\|_Q^2 \triangleq x^T Qx$. The function $\lceil x \rceil$ rounds x up to its nearest larger integer. We denote a sequence by $(1, 2, \cdots)$ and a set by $\{1, 2, \cdots\}$.

2 | PROBLEM FORMULATION

We consider a linear time-invariant system, where the controller is connected to the actuators through a network that is affected by correlated packet losses. The system is illustrated in Figure 1 and is of the form

$$\begin{aligned} x_{k+1} &= Ax_k + B\hat{u}_k + \omega_k \\ \hat{u}_k &= \rho_k u_k, \end{aligned} \tag{1}$$

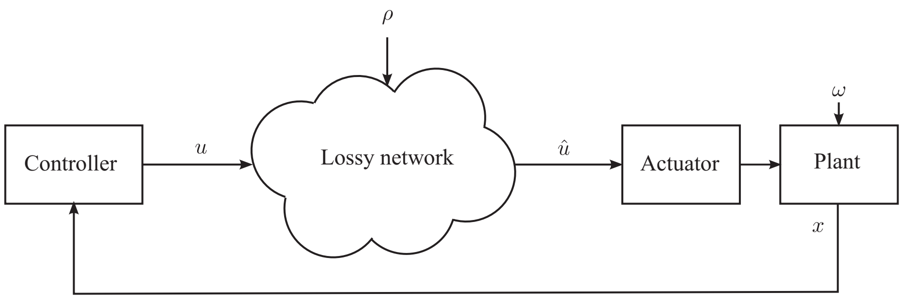

FIGURE 1 The system where the controller and the actuators are connected through a link that is affected by packet dropouts

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and $\omega_k \sim \mathcal{N}(0, \Sigma_\omega)$ is zero-mean white Gaussian noise with covariance Σ_ω. The binary variable $\rho_k \in \{0, 1\}$ indicates whether the packet at time k is received successfully at the actuator. Successful transmissions are acknowledged. In this work, we assume that the controller always receives the acknowledgements successfully.

Remark 1. In this work, we only consider packet dropouts in the actuator to controller link and assume that the feedback link (plant to controller) is deterministic, ie, no packet dropouts occur. However, we utilize acknowledgements. This means that the controller knows whether a packet is successfully received at the actuators or not. Therefore, the separation principle holds,[32] and in the optimal controller design that we present, the state can be replaced by its optimal estimate. Estimation for systems affected by packet losses in the feedback channel is considered in works, such as Smith and Seiler,[27] Schenato et al,[32] and Peters et al.[33]

The packet arrivals for the model in Equation 1 are random and correlated. This means that the probability for the packet at time k to arrive depends on the history of previously transmitted packets. In this work, we consider the case where the relevant length of the previous history is of a given length. More formally, we can state this as

$$\mathbf{Pr}\{\rho_k = 1 | \rho_{k-1}, \cdots, \rho_{k-\infty}\} = \mathbf{Pr}\{\rho_k = 1 | \rho_{k-1}, \cdots, \rho_{k-d}\},$$

where $0 \le d < \infty$ is the length of the relevant history. In the present setting, the controller always receives acknowledgements on the successful packet transmissions and, therefore, causally knows the exact packet arrival history. To simplify the notation, we accumulate the history of packet dropouts in the variable $\Theta_k \in \Xi = \left\{1, \cdots, 2^d\right\}$ to denote the binary sequence $(\rho_{k-1}, \cdots, \rho_{k-d})$, where $\Theta_k = i$ refers to a certain realization of $(\rho_{k-1}, \cdots, \rho_{k-d})$. Here, it is easy to see that Θ_k is first-order Markov, since

$$\mathbf{Pr}\{\Theta_{k+1} = j | \Theta_k = i, \Theta_{k-1}, \cdots, \Theta_0\} = \mathbf{Pr}\{\Theta_{k+1} = j | \Theta_k = i\} \triangleq p_{i,j}.$$

Furthermore, denote $\mathbf{Pr}\{\Theta_k = i\} \triangleq \pi_i$. We assign the outcomes of ρ_k to values of Θ_k by

$$\Theta_k = 1 + \sum_{\ell=1}^{d} \rho_{k-\ell} 2^{d-\ell}. \tag{2}$$

Note in Equation 2, that every possible binary sequence $(\rho_{k-1}, \cdots, \rho_{k-d})$ gets assigned to a unique value of Θ. Furthermore, note that

$$\rho_{k-1} = \begin{cases} 0, & \text{if} \quad \Theta_k \le r \\ 1, & \text{if} \quad \Theta_k > r, \end{cases} \tag{3}$$

where $r = 2^{d-1}$.

Since ρ_k is a binary variable, there are for each Θ_k only 2 possible outcomes of Θ_{k+1}: one where $\rho_k = 1$ and one where $\rho_k = 0$. To simplify the notation throughout the remainder of the paper, we introduce the variables ϕ_i and $\bar{\phi}_i$. Here, ϕ_i indicates the case when the packet arrival history at time k is given by $\Theta_k = i$, and the most recent packet is successfully received at the actuator ($\rho_k = 1$). Then, $\Theta_{k+1} = \phi_i$, where ϕ_i is computed as

$$\phi_i = \left\lceil \frac{i}{2} \right\rceil + 2^{d-1} \quad \in \left\{r + 1, \cdots, 2^d\right\}. \tag{4}$$

Likewise, when at time k, the packet arrival history is given by $\Theta_k = i$ and the packet transmission failed ($\rho_k = 0$), $\Theta_{k+1} = \bar{\phi}_i$, where

$$\bar{\phi}_i = \left\lceil \frac{i}{2} \right\rceil \quad \in \{1, \cdots, r\}. \tag{5}$$

It hereby follows that

$$\mathbf{Pr}\{\rho_k = 1 | \Theta_k = i\} = \mathbf{Pr}\{\Theta_{k+1} = \phi_i | \Theta_k = i\} = p_{i,\phi_i}$$

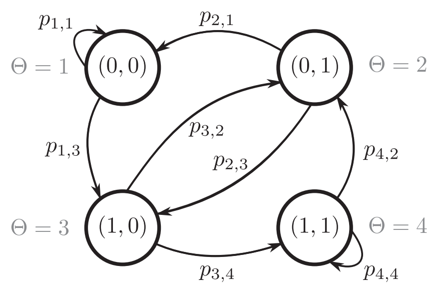

FIGURE 2 Illustration of the assignment of Θ to packet arrival sequences (ρ_{k-1}, ρ_{k-2}) by Equation 2. Here, the sequence $(1, 0)$ means that $\rho_{k-1} = 1$ and $\rho_{k-2} = 0$. The probability $p_{2,1} = \mathbf{Pr}\{\Theta_{k+1} = 1|\Theta_k = 2\} = \mathbf{Pr}\{\rho_k = 0|\rho_{k-1} = 0, \rho_{k-2} = 1\}$

and

$$\mathbf{Pr}\{\rho_k = 0|\Theta_k = i\} = \mathbf{Pr}\{\Theta_{k+1} = \bar{\phi}_i|\Theta_k = i\} = p_{i,\bar{\phi}_i} = 1 - p_{i,\phi_i}.$$

Example 1. Consider the case of a packet dropout distribution with $d = 2$. This means that the probability for a successful transmission at time k depends on the outcomes of ρ_{k-1} and ρ_{k-2}. This is illustrated in Figure 2. The Markov states $\Theta \in \Xi$, where $\Xi = \{1, 2, 3, 4\}$ are assigned using Equation 2. This can be presented using the Markov transition matrix

$$P = \begin{array}{c} \overbrace{\Theta = 1 \quad \Theta = 2}^{\rho_k = 0} \quad \overbrace{\Theta = 3 \quad \Theta = 4}^{\rho_k = 1} \\ \begin{bmatrix} p_{1,1} & 0 & p_{1,3} & 0 \\ p_{2,1} & 0 & p_{2,3} & 0 \\ 0 & p_{3,2} & 0 & p_{3,4} \\ 0 & p_{4,2} & 0 & p_{4,4} \end{bmatrix} \end{array} \begin{array}{l} \Theta = 1 \\ \Theta = 2 \\ \Theta = 3 \\ \Theta = 4, \end{array} \tag{6}$$

where $p_{1,3} = \mathbf{Pr}\{\rho_k = 1|\rho_{k-1} = 0, \rho_{k-2} = 0\}$. Here, $\phi_1 = 3$ and $\bar{\phi}_1 = 1$; thus, $p_{1,\phi_i} = p_{1,3}$. □

3 | OPTIMAL PREDICTIVE CONTROLLER DESIGN

We utilize predictive control to design a control law that minimizes a receding horizon cost function, which penalizes predicted future states and controls, as is done in LQ regulator (LQR) design. System (1) is however affected by random packet dropouts and noise, which are only known for the past, and not the current and future times. We therefore formulate the cost function using the expectation operator, where the expectation is taken over the packet dropout process ρ_k and the process noise ω_k. When minimizing the cost function using dynamic programming, we obtain feedback control policies as functions of the state and packet arrival history. The controller has at time k access to the state x_k and the Markov state Θ_k. We, thus, obtain controllers of the form

$$u_k = f_k(\Theta_k, x_k), \qquad k = 0, 1, \cdots, N-1.$$

We want to design the control law to minimize the receding horizon cost function

$$J_N\left(x_0, U_{0,N}, \Theta_0\right) = \mathbf{E}\left\{ \sum_{k=0}^{N-1} \|x_k\|_Q^2 + \rho_k \|u_k\|_R^2 + \|x_N\|_{S_{N,\Theta_N}}^2 \middle| x_0, \Theta_0 \right\}, \tag{7}$$

where $Q \geq 0$, $R \geq 0$, and $S_{N,i} \geq 0$, $\forall i \in \Xi$ are design parameters that trade off control performance vs control effort and the sequence $U_{k,N} \triangleq (u_k, \cdots, u_{N-1})$.

Remark 2. The problem that we presented here can be fitted into the general Markov jump linear system framework, see, eg, the work of Costa and Fragoso.[22] The difference between the framework presented in the work of Costa and Fragoso[22] and the current problem setting is that, we, at time k, do not have the outcome of ρ_k available. However, the same control laws can be obtained with both frameworks, ie, our framework and the framework presented in the work of Costa and Fragoso[22] by assuming that at time k, the packet transmission will always be successful.

In the rest of this section, the result that we present will differ from the work of Costa and Fragoso[22] on 2 points.

1. In the presented setting, it is a consequence of the problem formulation that the optimal strategy is to always design the control packet to assume that the current transmission is successful.

2. We take advantage of the special sparse structure of the transition matrix P, which results in simpler expressions, making it computationally more efficient. ■

Next, we present the controller design.

Proposition 1. *The optimal control at each time step is given by the linear function*

$$u_k^\star = L_{k,\Theta_k}^\star x_k, \tag{8}$$

where the control gain for each $i \in \Xi$ is given by

$$L_{k,i}^\star = -\left(R + B^T S_{k+1,\phi_i} B\right)^{-1} B^T S_{k+1,\phi_i} A \qquad \forall i \in \Xi, \tag{9}$$

with

$$S_{k,i} = Q + A^T\left(S_{k+1,\bar{\phi}_i} p_{i,\bar{\phi}_i} + S_{k+1,\phi_i} p_{i,\phi_i}\right) A - p_{i,\phi_i} A^T S_{k+1,\phi_i} B \left(R + B^T S_{k+1,\phi_i} B\right)^{-1} B^T S_{k+1,\phi_i} A, \tag{10}$$

and ϕ_i and $\bar{\phi}_i$ are defined in Equations 4 and 5. Here, $p_{i,\phi_i} = \mathbf{Pr}\{\Theta_{k+1} = \phi_i | \Theta_k = i\}$, and $S_{N,i}$ is given for all $i \in \Xi$. This results in the optimal cost

$$J_N^\star(x_0, \Theta_0) = x_0^T S_{0,\Theta_0} x_0 + c_{0,\Theta_0}, \tag{11}$$

where

$$c_{k,i} = \sum_{j=\{\phi_i, \bar{\phi}_i\}} \left(\text{trace}(\Sigma_\omega S_{k+1,j}) + c_{k+1,j}\right) p_{i,j} \tag{12}$$

with $c_{N,i} = 0$ for all i.

Proof. The proof is given in Appendix A. □

It is worth to note from Equation 9 that the control law at time k only depends on Θ_k for the computation of S_{k+1}, which, in turn, depends on Θ_{k+1}. Now, since the transition probabilities to go from Θ_k to Θ_{k+1} are not used in Equation 9, the value of $L_{k,i}$ will not depend on ρ_{k-d}. Therefore, only 2^{d-1} out of the 2^d control laws are unique and need to be stored in a lookup table.

Remark 3. Many network models, such as the Gilbert-Elliot model[13,14] (also referred to as the pq model), use a Markov chain with length $d = 1$. It is worth noting that in this case using Proposition 1, only 1 unique controller gain is obtained. This means that the controller here does not have to observe the history of the packet dropouts to find the appropriate control gain. ■

The results in Proposition 1 can be vectorized for a relevant history of length d by defining the matrix

$$\mathcal{V}_{i,j} \triangleq \begin{bmatrix} I_n\sqrt{p_{(i+1),(j+1)}} & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & I_n\sqrt{p_{(i+3),(j+2)}} & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & I_n\sqrt{p_{(i+2^d-1),(j+r)}} \end{bmatrix} \in \mathbb{R}^{(2^d-1)n \times rn}$$

and

$$\mathcal{W}_{0,j} \triangleq \begin{bmatrix} \mathcal{V}_{0,j} \\ \mathbf{0}_{n,rn} \end{bmatrix} \in \mathbb{R}^{2^d n \times rn} \qquad \mathcal{W}_{1,j} \triangleq \begin{bmatrix} \mathbf{0}_{n,rn} \\ \mathcal{V}_{1,j} \end{bmatrix} \in \mathbb{R}^{2^d n \times rn}.$$

Now, define the following matrices describing the successful and lost transmissions:

$$\mathcal{W} \triangleq \left[\, \mathbf{0}_{2^d n \times rn} \ \mathcal{W}_{0,r} \ \mathbf{0}_{2^d n \times rn} \ \mathcal{W}_{1,r} \,\right] \in \mathbb{R}^{2^d n \times 2^{d+1} n}$$
$$\bar{\mathcal{W}} \triangleq \left[\, \mathcal{W}_{0,0} \ \mathbf{0}_{2^d n \times rn} \ \mathcal{W}_{1,0} \ \mathbf{0}_{2^d n \times rn} \,\right] \in \mathbb{R}^{2^d n \times 2^{d+1} n},$$

respectively. Define $\bar{Q} \triangleq \text{diag}\{Q\}$, $\bar{A} \triangleq \text{diag}\{A\}$, $\bar{B} \triangleq \text{diag}\{B\}$. Furthermore, let

$$\bar{R} \triangleq \text{diag}\left\{\left[R p_{1,\phi_1}, R p_{2,\phi_2}, \cdots, R p_{d,\phi_d}\right]\right\}$$
$$\bar{S}_k \triangleq \text{diag}\left\{\left[S_{k,1}, S_{k,2}, \cdots, S_{k,2^d}\right]\right\},$$

and

$$\Gamma_k \triangleq \mathcal{W} \hat{S}_k \mathcal{W}^T \tag{13}$$

$$\bar{\Gamma}_k \triangleq \bar{\mathcal{W}} \hat{S}_k \bar{\mathcal{W}}^T, \tag{14}$$

where

$$\hat{S}_k \triangleq I_2 \otimes \bar{S}_k.$$

Then, Equation 10 can be expressed as

$$\bar{S}_k = \bar{Q} + \bar{A}^T\left(\Gamma_{k+1} + \bar{\Gamma}_{k+1}\right)\bar{A} - \bar{A}^T\Gamma_{k+1}\bar{B}\left(\bar{R} + \bar{B}^T\Gamma_{k+1}\bar{B}\right)^{-1}\bar{B}^T\Gamma_{k+1}\bar{A}. \tag{15}$$

The augmented control gains can then be computed by

$$\mathcal{L}_k^\star = \left(\bar{R} + \bar{B}^T\Gamma_{k+1}\bar{B}\right)^{-1}\bar{B}^T\Gamma_{k+1}\bar{A}, \tag{16}$$

where $\mathcal{L}_k^\star$ is a block diagonal matrix with the control gains $L_{k,i}^\star$, $i \in \Xi$ from Equation 9 presented on the diagonal. The control gain $L_{k,i}^\star$ can be extracted as the ith $n \times m$ diagonal block of $\mathcal{L}_k^\star$.

To implement the predictive controller design from Proposition 1, one would use a lookup table that contains the control gains, and then at each time step, use the history that is contained in Θ_k to select the corresponding control gain. This requires enough memory to store the 2^{d-1} control gains. One question that arises here is whether it is necessary to take the full packet dropout history into account in the controller design to guarantee a reasonable control performance. Omitting part of the history will result in fewer control laws, and thereby a smaller lookup table. In the next section, we will present suboptimal controller designs that reduce the amount of unique control laws at the cost of performance.

4 | SUBOPTIMAL REDUCED COMPLEXITY CONTROLLERS

When designing the optimal control law for system (1) affected by correlated packet losses, we, as described in Section 3, obtain a controller for every possible sequence of packet arrivals. This, for a relevant history of length d, results in 2^{d-1} unique controllers. The amount of controllers will thus grow exponentially when a longer relevant history is considered. In the remainder of this work, we will study methods to reduce the number of controllers. This, however, will inevitably result in some degradation in the control performance. We will therefore investigate different methods that reduce the number of controllers while maintaining satisfying control performance.

We propose to reduce the amount of controllers by designing a controller for a group of possible packet arrival sequences. This, for example, means that we design 1 controller that is to be applied for packet arrival sequences $\Theta = \{1, 2, 3, 4\}$. This, however, leads to 2 questions:

1. How to group the Markov states that govern the different packet arrival sequences, and
2. what controller to use for each of these groups?

This problem thus breaks down to a grouping and a controller design problem. First, we will present a method on how to group the packet arrival sequences. Then, we present 3 possible methods for the controller design. The 3 controller design methods differ in where in the process the grouping, and thereby reduction (or quantization) is done. This is illustrated graphically in Figure 3. The 3 methods are as follows:

- **Markov averaged (MA)**: The grouping is done for the Markov chain. Here, we reduce or aggregate the Markov chain prior to the controller design. The controller design is thus done for the reduced Markov chain, which therefore will lead to a reduced number of controllers.
- **Group averaged (GA)**: The grouping is done after the controller design. The controller design is thus done for the full Markov chain, and afterward, the number of controllers is reduced.
- **Cost averaged (CA)**: In this method, the controller design is done for the full Markov chain, and the reduction of the number of controllers is enforced in the design.

In the MA design method, the quantization is done when reducing the Markov chain prior to the controller design. This thus means that the optimal controller is designed for a less precise network model, which will result in performance loss. In the GA controller design, one first obtains the full complexity optimal controllers for the full Markov chain. To reduce the number of controllers, the quantization is applied at the end of the design process. This thus means that the suboptimality is first introduced at the end of the design process. Both of these methods utilize the controller design that is presented in Proposition 1 and are relatively simple to implement. Finally, in the CA design method, we enforce the quantization when the controller design is performed. Here, the suboptimality is taken into account in the cost function directly. The control laws that are obtained here should therefore result in a lower cost, and therefore also better performance, than the controllers that are obtained using the MA and GA methods. However, the CA controller design is more complex.

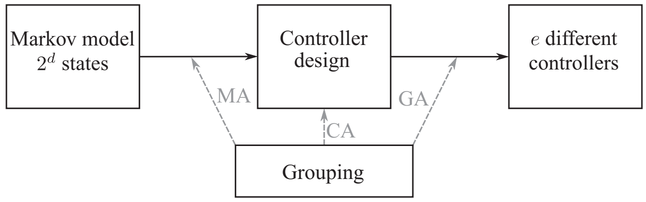

FIGURE 3 An illustration of where to do the grouping in the controller design. Here, one can reduce the number of Markov states prior to the controller design (Markov averaged [MA]), reduce the number of control laws after the controller design (group averaged [GA]), or reduce the number of control laws while doing the controller design (cost averaged [CA])

Next, in Section 4.1, we present a method on how to group the packet arrival sequences. This grouping method is applied in all 3 reduced complexity controller designs. The difference between the controller design methods is, as mentioned before, where in the design process the grouping function is applied. After presenting the grouping method, we present the MA, GA, and CA designs in Sections 4.2 to 4.4.

4.1 | Grouping the packet arrival sequences

In this section, we will discuss how to group the packet arrival sequences, such that we then can design a controller for each of these groups. In Section 2, we assigned the packet arrival sequences to values $\Theta \in \Xi$ using Equation 2. In this section, we present a surjective function $g : \Xi \to \hat{\Xi}$, which maps the individual packet arrival sequences to a group, for which we then design a controller for. Since we aim at reducing the number of controllers, card $\left(\hat{\Xi}\right) \leq$ card (Ξ). Let $\hat{\Theta} \in \hat{\Xi}$ indicate the number of the group and associate with $\hat{\Theta} = v$ the subset $\hat{\Xi}_v$, which is defined as

$$\hat{\Xi}_v \triangleq \{i \in \Xi : g(i) = v\}, \tag{17}$$

to indicate which Θ get mapped to $\hat{\Theta} = v$. Note that $\hat{\Xi}_v$ are disjoint sets ($\hat{\Xi}_v \cap \hat{\Xi}_j = \emptyset$ when $v \neq j$).

The design of the grouping function is a well-known problem.[34,35] The only method on how to find the optimal grouping function is by performing a combinatorial search among all the possible grouping functions, which is a non-deterministic polynomial-time (NP)-hard problem. To avoid this, we instead propose to design the grouping function g, such that the packet arrival sequences are grouped by their most recent packet arrival history. The intuition behind this is that the most recent history has the largest impact on the outcomes of future packet dropouts. Thus, we group the packet arrival sequences such that the most recent packet arrival history $\left(\rho_{k-1}, \rho_{k-2}, \cdots, \rho_{d-q}\right)$, where $0 \leq q \leq d$, is identical for all $\Theta \in \hat{\Xi}_v$. This results in a total of 2^{d-q} groups. The grouping function g is in this case given by

$$g(\Theta) = \left\lceil \frac{\Theta}{2^q} \right\rceil, \tag{18}$$

and let $\hat{\Theta} = g(\Theta)$. We then design control laws for each group, such that these become

$$\hat{u}_k = \hat{f}_k\left(\hat{\Theta}_k, x_k\right), \tag{19}$$

where variable $\hat{\Theta}_k \in \hat{\Xi}$ with $\hat{\Xi} = \left\{1, \cdots, 2^{d-q}\right\}$, and each $\hat{\Theta}_k = i$ is linked to a given packet arrival sequence of the most recent $d - q$ packet arrivals.

Example 2. Consider the case from Example 1. Then, with $q = 1$, we group the packet arrival sequences by the most recent transmission outcome. Thus, $\hat{\Theta}$ takes values in $\hat{\Xi} = \{1, 2\}$. This is illustrated in Figure 4. Using Equations 17 and 18, we obtain that $\hat{\Xi}_1 = (1, 2)$ and $\hat{\Xi}_2 = (3, 4)$. □

4.2 | Optimal controller for reduced Markov chain (MA)

In this section, we present the MA controller design. We here use the grouping function g from Equation 18 to obtain the groups of packet arrival sequences as explained in Section 4.1. We then compute the transition probabilities between the groups $\hat{\Theta} \in \hat{\Xi}$ from the transition probabilities of the full packet arrival sequences. This is called the aggregation of the Markov chain and is a well-studied topic for efficient steady-state analysis of large Markov chains (see the work of Schweitzer[35] and the references therein). We hereby obtain a reduced order Markov chain with transition matrix $\hat{P} \in \mathbb{R}^{2^{d-q} \times 2^{d-q}}$. For a given state space for the

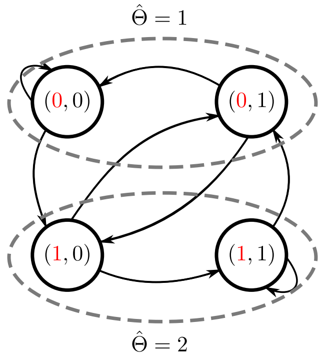

FIGURE 4 An example of the grouping of the packet arrival sequences. Here, $q = 1$ and the relevant history in each group (encircled by the dashed line) is marked in red [Colour figure can be viewed at wileyonlinelibrary.com]

network model, and a choice of grouping of the reduced state space, the aggregate steady-state probabilities are for each group computed by[35]

$$\bar{\pi}_v = \mathbf{Pr}\left\{\hat{\Theta} = v\right\} = \sum_{j \in \hat{\Xi}_v} \pi_j, \quad \forall v \in \hat{\Xi},$$

and aggregate transition probabilities by

$$\hat{p}_{v,\ell} = \mathbf{Pr}\left\{\hat{\Theta}_k = \ell \middle| \hat{\Theta}_{k-1} = v\right\} = \frac{\sum_{j \in \hat{\Xi}_v} \sum_{i \in \hat{\Xi}_\ell} \pi_i p_{i,j}}{\sum_{j \in \hat{\Xi}_v} \pi_j},$$

which forms the aggregated transition matrix $\hat{P}$.

Afterwards, we design an optimal control law using Proposition 1 for the reduced Markov chain with the transition matrix $\hat{P}$. This results in a reduced number of controllers. Since we reduce, or quantize, the Markov model prior to the controller design, we obtain an optimal control law for a Markov model that is representative, but not completely describing the actual network model that is affecting the controlled system.

4.3 | GA controller

The other simple approach that we present for the reduced complexity controller is the GA controller. With this method, we first design the optimal controllers for the entire Markov chain in the state-space Ξ using Proposition 1. We then, using the grouping function g from Equation 18 in Section 4.1, group the controllers, such that there is 1 controller for every group $\hat{\Theta}$ of packet arrival sequences. The reduced controllers will then be of the form

$$\hat{u}_k = \hat{L}_{k,v} x_k \qquad g(\Theta_k) = v,$$

where the control gains are given by

$$\begin{aligned} \hat{L}_{k,v} &= \mathbf{E}\left\{L_k \middle| \hat{\Theta}_k = v\right\}, \quad \forall v \in \hat{\Xi} \\ &= \sum_{j \in \hat{\Xi}_i} L_{k,j} \frac{\pi_j}{\hat{\pi}_v}, \end{aligned} \tag{20}$$

with

$$\hat{\pi}_v \triangleq \mathbf{Pr}\left\{\hat{\Theta}_k = v\right\} = \sum_{j \in \hat{\Xi}_v} \pi_j.$$

4.4 | Reducing the amount of controllers in the design (CA)

In this section, we design the CA controller. In this method, we force the control gain to be equal for all possible packet arrival sequences that are contained in a group $\hat{\Theta}$, where the grouping is done as described in Section 4.1. The control law is thus designed by taking the expected value over the full history of ρ in the cost function and then setting the controller for the different outcomes of Θ to be identical. The difference to the MA design is here, that we, in the MA design, reduce the Markov chain prior to the controller design. We thereby introduce suboptimality into the design process before designing the controllers. On the contrary, in the CA controller, we take the full Markov chain into account while minimizing the cost, but then introduce the suboptimality directly when minimizing the cost function by forcing some of the controllers to be equal.

Here, we aim at forcing the grouping of the packet arrival sequences through in the cost function that we utilize to design the predictive controller. This is done by designing a control law of the form

$$\hat{u}_{k,i} = \hat{f}_k\left(g(i), x_k\right), \quad \forall i \in \hat{\Xi} \tag{21}$$

to minimize the finite-horizon cost function

$$\begin{aligned} \bar{J}_N\left(U_{0,N}, x_0\right) &= \mathbf{E}\left\{J_N\left(U_{0,N}, x_0, \Theta_0\right) | x_0\right\} \\ &= \mathbf{E}\left\{\sum_{k=0}^{N-1} \|x_k\|_Q^2 + \rho_k \|u_k\|_R^2 + \|x_N\|_{S_{N,\Theta_N}}^2 \middle| x_0 \right\}. \end{aligned} \tag{22}$$

Since g maps Ξ onto $\hat{\Xi}$, the controller $\hat{f}_k\left(g(i), x_k\right)$ is identical for all $i \in \Xi$ that $g(i)$ maps to the same $\hat{\Theta} \in \hat{\Xi}$. We, therefore, obtain the sequence of reduced controllers $\bar{U}_{0,N} = (\bar{u}_0, \cdots, \bar{u}_{N-1})$ with $\bar{u}_k = \left(\hat{u}_{k,1}, \cdots, \hat{u}_{k,2^{d-q}}\right)$.

The controller design is thus done by evaluating the optimization problem

$$\min_{U_{0,N}} \quad \bar{J}_N\left(U_{0,N}, x_0\right) \tag{23a}$$

$$\text{subject to} \quad u_{k,i} = \hat{f}_k\left(g(i), x_k\right) \qquad i \in \Xi, k = 0, \cdots, N-1. \tag{23b}$$

Furthermore, note that the control law (21) is only allowed to depend on x_k and through $g(\Theta_k)$ on $\hat{\Theta}_k$, and thereby not on previously applied controls. This allows us to obtain closed-form solutions to problem (23a) and (23b). Note, however, that these control laws not necessarily are optimal, as described in the works of Yang et al[36] and Vargas et al.[37,38] The control laws that are presented in the works of Vargas et al[37,38] cannot, however, be computed using closed-form expressions, and an iterative algorithm is required. We, therefore, instead focus on suboptimal control laws that allow us to obtain closed-form expressions. We obtain the following result.

Theorem 1. *The controls of form (21) for Equations 23a and 23b are given by*

$$\hat{u}_k^\star = \bar{L}_{k,v}^\star x_k, \quad g(\Theta_k) = v, \tag{24}$$

where

$$\bar{L}_{k,v}^\star = -\left(\bar{R}_v + B^T \bar{S}_{k+1,v} B\right)^{-1} B^T \bar{S}_{k+1,v} A \tag{25}$$

and for all $i \in \Xi$

$$S_{k,i} = Q + A^T S_{k+1,\bar{\phi}_i} A p_{i,\bar{\phi}_i} + \left(\bar{L}_{k,g(i)}^\star\right)^T R \bar{L}_{k,g(i)}^\star p_{i,\phi_i} + \left(A + B\bar{L}_{k,g(i)}^\star\right)^T S_{k+1,\phi_i}\left(A + B\bar{L}_{k,g(i)}^\star\right) p_{i,\phi_i}. \tag{26}$$

Here, $S_{N,i} = S_N$ and

$$\bar{S}_{k,v} = \sum_{i \in \hat{\Xi}_v} S_{k,\phi_i} p_{i,\phi_i} \pi_i \tag{27}$$

$$\bar{R}_v = R \sum_{i \in \hat{\Xi}_v} p_{i,\phi_i} \pi_i. \tag{28}$$

This results in the cost

$$\bar{J}_N^\star(x_0) = \sum_{i \in \Xi} x_0^T S_{0,i} x_0 \pi_i + c_{0,i} \pi_i, \tag{29}$$

where

$$c_{k,i} = \operatorname{trace}\left(\Sigma_\omega\left(S_{k+1,\phi_i} p_{i,\phi_i} + S_{k+1,\bar{\phi}_i} p_{i,\bar{\phi}_i}\right)\right) + c_{k+1,\phi_i} p_{i,\phi_i} + c_{k+1,\bar{\phi}_i} p_{i,\bar{\phi}_i}, \tag{30}$$

with $c_{N,i} = 0$, $i \in \Xi$.

Proof. The proof of Theorem 1 is presented in Appendix B. □

TABLE 1 The conditional probabilities for the network model

$(\rho_{k-1}, \cdots, \rho_{k-4})$	Θ	$p_{i,\bar{\phi}(i)}$ and $\Pr\{\rho_k = 0\}$	$p_{i,\phi(i)}$ and $\Pr\{\rho_k = 1\}$
0000	1	$p_{1,1} = 0.2$	$p_{1,9} = 0.8$
0001	2	$p_{2,1} = 0.65$	$p_{2,9} = 0.35$
0010	3	$p_{3,2} = 0.78$	$p_{3,10} = 0.22$
0011	4	$p_{4,2} = 0.2$	$p_{4,10} = 0.8$
0100	5	$p_{5,3} = 0.85$	$p_{5,11} = 0.15$
0101	6	$p_{6,3} = 0.3$	$p_{6,11} = 0.7$
0110	7	$p_{7,4} = 0.75$	$p_{7,12} = 0.25$
0111	8	$p_{8,4} = 0.2$	$p_{8,12} = 0.8$
1000	9	$p_{9,5} = 0.6$	$p_{9,13} = 0.4$
1001	10	$p_{10,5} = 0.8$	$p_{10,13} = 0.2$
1010	11	$p_{11,6} = 0.5$	$p_{11,14} = 0.5$
1011	12	$p_{12,6} = 0.3$	$p_{12,14} = 0.7$
1100	13	$p_{13,7} = 0.5$	$p_{13,15} = 0.5$
1101	14	$p_{14,7} = 0.1$	$p_{14,15} = 0.9$
1110	15	$p_{15,8} = 0.4$	$p_{15,16} = 0.6$
1111	16	$p_{16,8} = 0.7$	$p_{16,16} = 0.3$

5 | SIMULATION STUDIES

We compare the performance of the predictive controller designs proposed in Sections 3 and 4 using Monte Carlo simulation studies. The controllers are implemented as model predictive control, that is, we set $x_k = L_{0,\Theta_k} x_k$ for all k. We demonstrate the performance of the proposed controller designs for both a direct-current (DC) servo and a mass-spring system with 3 masses. In both systems, consider a relevant history of length $d = 4$. This results in 16 possible packet arrival sequences and thereby 8 unique controllers. These packet arrival sequences are assigned to Markov states using Equation 2. The conditional transmission success and failure probabilities are given in Table 1. Both simulations consider a system of the form (1) with $\Sigma_\omega = I_m$, and the results are obtained by averaging 500 simulations each of length 50 000. The costs are computed with horizon length $N = 500$ to obtain a steady-state solution. A standard LQR controller that does not take the network conditions into account is used to benchmark. We also include a controller that considers utilizing an *i.i.d* model of the network. The *i.i.d* packet dropout is computed from the steady-state probabilities of the Markov chain in Table 1. This results in a probability of $\mathbf{Pr}\{\gamma_k = 1\} = 0.5308$ for a successful transmission. The controller design for the *i.i.d* model is done, as described in the work of Peters et al.[39] The controller performance is measured using the empirical cost

$$\text{cost} = \frac{1}{50\,000} \sum_{k=1}^{50\,000} x_k^T Q x_k + \rho_k u_k^T R u_k.$$

5.1 | DC servo

Here, we consider a DC servo with states $x = \begin{bmatrix} \dot{\Theta} & \Theta \end{bmatrix}^T$ with Θ being the angular position and $\dot{\Theta}$ the angular velocity. The continuous-time model is given by

$$A = \begin{bmatrix} \frac{1}{\tau} & 0 \\ 1 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} \frac{K}{\tau} \\ 0 \end{bmatrix}.$$

We select time constant $\tau = 5$ and gain $K = 50$. The system has been discretized using `c2d` in MATLAB R2017a with a sampling interval of 1 second, resulting in the discrete dynamics

$$A = \begin{bmatrix} 0.819 & 0 \\ 0.906 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 9.063 \\ 4.683 \end{bmatrix}$$

and has open-loop eigenvalues located at 1 and 0.819.

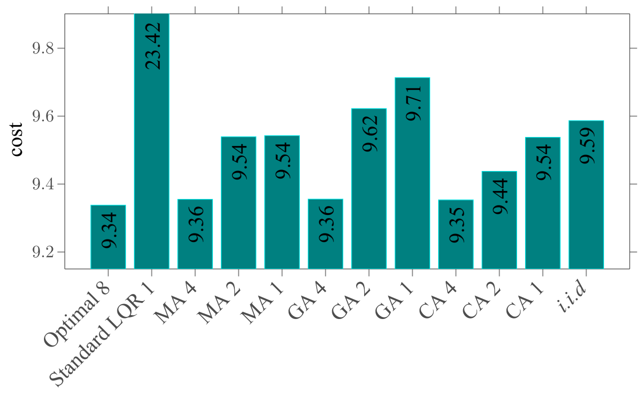

FIGURE 5 Averaged results for the simulations of the direct-current (DC) servo. The optimal control law is designed using Proposition 1, the Markov-averaged (MA), group-averaged (GA), and cost-averaged (CA) control laws, designed using Sections 4.2 and 4.3 and Theorem 1, respectively. The number trialing the name indicates the amount of unique controllers [Colour figure can be viewed at wileyonlinelibrary.com]

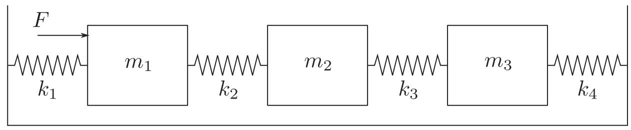

FIGURE 6 An illustration of the model with 3 masses connected by springs that is considered in the simulation in Section 5.2. The actuator exerts force F on the first mass

The simulation results in Figure 5 show that the stochastic controller design (Optimal 8) presented in Proposition 1 significantly outperforms LQR, which results in a cost of 23.4. Compared with the *i.i.d* controller design from the work of Peters et al,[39] the cost is reduced by 2.61%.

When reducing the number of controllers to 4 controllers, the performance is only slightly reduced compared with the optimal design. However, when only 2 controllers are desired, the CA controller (from Theorem 1) results in the lowest performance reduction, followed by the MA design (Section 4.2). Interestingly, the performance of the MA design is identical when 1 or 2 controllers are used. This matches the performance of the CA design with 1 controller. Both the CA and MA controllers outperform the *i.i.d* controller. The GA controller, presented in Section 4.3, does not lose much performance when only 4 control laws are used. Reducing the number of controllers to 1 or 2, however, results in noticeably poorer performance than the *i.i.d* design.

5.2 | Mass-spring system

In this simulation, we consider a system with 3 masses on a surface connected by springs to each other and sidewalls. This system is depicted in Figure 6. There is an actuator exerting a force to the first mass. There is an integrator at the actuator such that the previous control input is held at the actuator. The masses are denoted by m and are equal. The spring coefficients are denoted by k and are also all equal. This system has continuous-time dynamics given by

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -2k_m & k_m & 0 & 0 & 0 & 0 & 1 \\ k_m & -2k_m & k_m & 0 & 0 & 0 & 0 \\ 0 & k_m & -2k_m & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{m} \\ 0 \\ 0 \\ \frac{1}{m} \end{bmatrix},$$

where $k_m = \frac{k}{m}$. We set $m = 1$ and $k = 0.323$. This gives open-loop eigenvalues at $\pm 1.05j$, $\pm 0.804j$, $\pm 0.435j$, and one at 0 due to the integrator. We discretize the system using `c2d` in MATLAB R2017a with a sampling interval of 1 second. The discrete-time system has eigenvalues at $0.498 \pm 0.867j$, $0.694 \pm 0.72j$, $0.907 \pm 0.421j$, and 1.

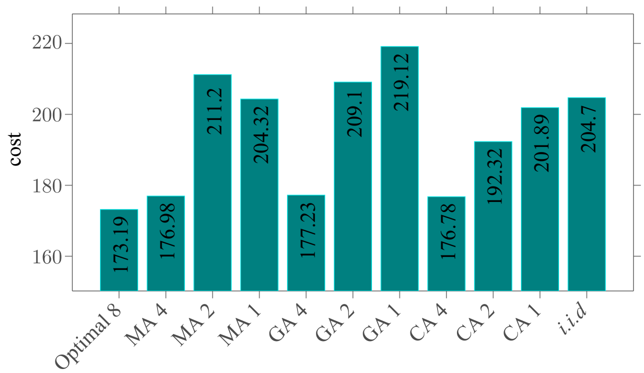

FIGURE 7 Averaged results for the simulations of the mass-spring system. The optimal control law designed using Proposition 1, the Markov-averaged (MA), group-averaged (GA), and cost-averaged (CA) control laws, designed using Sections 4.2 and 4.3 and Theorem 1, respectively. The number trialing the name indicates the amount of unique controllers [Colour figure can be viewed at wileyonlinelibrary.com]

The simulation results are shown in Figure 7. The standard LQR did not manage to maintain stability and is therefore omitted. The optimal controller that takes the full history into account (Optimal 8) performs 18.2% better than the *i.i.d* controller from the work of Peters et al.[39] When the number of controllers is decreased, the control performance is reduced. Similar to the simulations for the DC servo, the CA controller outperformed the MA and GA controllers. When only one control law is desired, the MA and CA controllers outperform the *i.i.d* controller by a small margin. It is, however, noteworthy that with the MA with only 1 controller, the performance is better than the MA with 2 controllers. This is believed to be due to the combination of a suboptimal selection of the grouping function and a suboptimality that is introduced when the Markov chain is reduced. The CA method outperforms the other methods; however, with only 1 controller, the performance difference between the CA and MA is small.

6 | CONCLUSIONS

We studied stochastic predictive control for systems that are affected by correlated packet dropouts between the controller and the actuators. The probability for a successful transmission depends on a history of packet transmission outcomes. This results in a special structure for the Markov transition matrix, which we take advantage of to obtain simple expressions for the controller design. We obtain feedback policies that take into account the packet arrival history by selecting representative gains from a lookup table. Simulation studies show that the designed controller outperforms the standard LQR controller design by a large margin, and for some systems, the presented designs are able to maintain stability where the traditional LQR fails. An implementation of this controller will, however, result in a significantly sized lookup table.

To reduce the size of the lookup table for the controllers, we presented 3 suboptimal methods to design the control laws. The difference between these methods is where the reduction of the amount of controllers is performed. All 3 of these methods show merely a minor performance degradation when the number of controllers is reduced, which allows for a good trade-off between performance and controller complexity. From the 3 algorithms, the best results for the reduced complexity controller design are obtained using the CA controller, which enforces the reduction of the number of controllers through the cost. It is also noteworthy that while the design procedure of the GA controller is simple, there is, in general, only little performance loss compared with the CA controller.

ORCID

Edwin G.W. Peters http://orcid.org/0000-0003-3766-8576
Daniel E. Quevedo http://orcid.org/0000-0002-4804-5481

REFERENCES

1. Reed BL, Hover FS. JLS-PPC: A jump linear system framework for networked control. *IEEE Trans Control Syst Technol*. 2017;25(3):924-939.
2. Lin H, Su H, Shu Z, Wu ZG, Xu Y. Optimal estimation in udp-like networked control systems with intermittent inputs: stability analysis and suboptimal filter design. *IEEE Trans Autom Control*. 2016;61(7):1794-1809.

3. Quevedo DE, Silva EI, Goodwin GC. Control over unreliable networks affected by packet erasures and variable transmission delays. *IEEE J Sel Areas Commun*. 2008;26(4):672-685.
4. Quevedo DE, Østergaard J, Nesic D. Packetized predictive control of stochastic systems over bit-rate limited channels with packet loss. *IEEE Trans Autom Control*. 2011;56(12):2854-2868.
5. Imer OC, Yüksel S, Başar T. Optimal control of LTI systems over unreliable communication links. *Automatica*. 2006;42(9):1429-1439.
6. Wu J, Chen T. Design of networked control systems with packet dropouts. *IEEE Trans Autom Control*. 2007;52(7):1314-1319.
7. Xia Y, Xie W, Zhu Z, Wang G, Wang X. Performance analysis of networked predictive control systems with data dropout. *Optim Control Appl Methods*. 2013;34(6):742-756.
8. Zhang H, Shi Y, Wang J. Observer-based tracking controller design for networked predictive control systems with uncertain Markov delays. *Int J Control*. 2013;86(10):1824-1836.
9. Zhang L, Shi Y, Chen T, Huang B. A new method for stabilization of networked control systems with random delays. *IEEE Trans Autom Control*. 2005;50(8):1177-1181.
10. Quevedo DE, Johansson KH, Ahlén A, Jurado I. Adaptive controller placement for wireless sensoractuator networks with erasure channels. *Automatica*. 2013;49(11):3458-3466.
11. Zhang D, Shi P, Wang QG, Yu L. Analysis and synthesis of networked control systems: a survey of recent advances and challenges. *ISA Trans*. 2017;66:376-392.
12. Konrad A, Zhao BY, Joseph AD, Ludwig R. A Markov-based channel model algorithm for wireless networks. *Wirel Netw*. 2003;9(3):189-199.
13. Gilbert EN. Capacity of a burst-noise channel. *Bell Syst Tech J*. 1960;39(5):1253-1265.
14. Lindbom L, Ahlén A, Sternad M, Falkenström M. Tracking of time-varying mobile radio channels part II: a case study on d-AMPS 1900 channels. *IEEE Trans Commun*. 2000;49:2207-2217.
15. Peters EGW, Quevedo DE, Østergaard J. Shaped Gaussian dictionaries for quantized networked control systems with correlated dropouts. *IEEE Trans Signal Process*. 2016;64(1):203-213.
16. Chen W, Zou Y, Xiao N, Song J, Niu Y. Quantised feedback stabilisation of LTI systems over partly unknown Markov fading channels. *Int J Syst Sci*. 2017;0(0):1-9.
17. Mo Y, Garone E, Sinopoli B. LQG control with Markovian packet loss. Paper presented at: 2013 European Control Conference (ECC); 2013; Zürich.
18. Song H, Chen SC, Yam Y. Sliding mode control for discrete-time systems with Markovian packet dropouts. *IEEE Trans Cybern*. 2016:1-11.
19. Sadeghi P, Kennedy RA, Rapajic PB, Shams R. Finite-state Markov modeling of fading channels – a survey of principles and applications. *IEEE Signal Process Mag*. 2008;25(5):57-80.
20. Wang HS, Moayeri N. Finite-state Markov channel–a useful model for radio communication channels. *IEEE Trans Veh Technol*. 1995;44(1):163-171.
21. Yajnik M, Kurose J, Towsley D. Packet loss correlation in the MBone multicast network. Paper presented at: Global Telecommunications Conference, 1996. GLOBECOM 96 Communications: The Key to Global Prosperity; 1996; IEEE, London.
22. Costa OLV, Fragoso MD, Marques RP. *Discrete-Time Markov Jump Linear Systems*. New York, NY, USA: Springer Science & Business Media: 2006.
23. Fragoso MD. Discrete-time jump LQG problem. *Int J Syst Sci*. 1989;20(12):2539-2545.
24. Casiello F, Loparo KA. Optimal control of unknown parameter systems. *IEEE Trans Autom Control*. 1989;34(10):1092-1094.
25. Val JBRD, Başar T. Receding horizon control of Markov jump linear systems. Paper presented at: Proceedings of the 1997, American Control Conference, vol. 5; 1997; Albuquerque.
26. Dolz D, Quevedo D, Peñarrocha I, Sanchis R, et al. Performance vs complexity trade-offs for Markovian networked jump estimators. Paper presented at: Proceedings of the 19th IFAC World Congress, 2014. International Federation of Automatic Control (IFAC); 2014; Cape Town.
27. Smith S, Seiler P. Estimation with lossy measurements: jump estimators for jump systems. *IEEE Trans Autom Control*. 2003;48(12):2163-2171.
28. Shi Y, Fang H. Kalman filter-based identification for systems with randomly missing measurements in a network environment. *Int J Control*. 2010;83(3):538-551.
29. Wang Z, Yang F, Ho DWC, Liu X. Robust H_∞ control for networked systems with random packet losses. *IEEE Trans Syst Man Cybern, Part B (Cybernetics)*. 2007;37(4):916-924.
30. Quevedo DE, Nešić D. Robust stability of packetized predictive control of nonlinear systems with disturbances and Markovian packet losses. *Automatica*. 2012;48(8):1803-1811.
31. Quevedo D, Nešić D. Input-to-state stability of packetized predictive control over unreliable networks affected by packet-dropouts. *IEEE Trans Autom Control*. 2011;56(2):370-375.
32. Schenato L, Sinopoli B, Franceschetti M, Poolla K, Sastry S. Foundations of control and estimation over lossy networks. *Proc IEEE*. 2007;95(1):163-187.
33. Peters EGW, Quevedo DE, Fu M. Co-design for control and scheduling over wireless industrial control networks. Paper presented at: 54th IEEE Conference on Decision and Control, CDC 2015, vol. 2015; December 15-18, 2015; Osaka, Japan.
34. Deng K, Mehta PG, Meyn SP. Optimal Kullback-Leibler aggregation via spectral theory of Markov chains. *IEEE Trans Autom Control*. 2011;56(12):2793-2808.
35. Schweitzer PJ. A survey of aggregation-disaggregation in large Markov chains. In: Stewart W, ed. *Numerical Solution of Markov Chains*, Chap. 4, vol. 8, Boca Raton, Florida, USA: CRC press; 1991:63-88.

36. Yang C, Bar-Shalom Y, Lin CF. Control of partially observed discrete-time jump systems. Paper presented at: American Control Conference, 1991; 1991; Massachusetts.
37. Vargas AN, do Val JBR, Costa EF. Receding horizon control of Markov jump linear systems subject to noise and unobserved state chain. Paper presented at: 43rd IEEE Conference on Decision and Control, CDC 2004, vol. 4; 2004; Paradise Island.
38. Vargas AN, do Val JBR, Costa EF. Optimality condition for the receding horizon control of Markov jump linear systems with non-observed chain and linear feedback controls. Paper presented at: Proceedings of the 44th IEEE Conference on Decision and Control; 2005; Seville.
39. Peters EGW, Marelli D, Fu M, Quevedo DE. Unified approach to controller and MMSE estimator design with intermittent communications. Paper presented at: 55th IEEE Conference on Decision and Control, CDC. December 12-14, 2016; Las Vegas, NV, USA.
40. Bertsekas DP. *Dynamic Programming and Optimal Control*. 3rd ed. Vol. 1. Nashua, NH, USA: Athena Scientific; 2005.

How to cite this article: Peters EGW, Marelli D, Quevedo DE, Fu M. Predictive control for networked systems affected by correlated packet loss. *Int J Robust Nonlinear Control*. 2017;1-17. https://doi.org/10.1002/rnc.3896

APPENDIX A

PROOF OF PROPOSITION 1

This proof is done using dynamic programming.[40] We define the optimal cost

$$J_N^\star(x_0,\Theta_0)=\min_{U_{0,N-1}} J_N\left(U_{0,N-1},x_0,\Theta_0\right).$$

Note that the cost at any stage n can be written as

$$J_N^\star(x_0,\Theta_0)=\min_{(u_0,\cdots,u_n)}\mathbf{E}\left\{\sum_{k=0}^{n}\|x_k\|_Q^2+\rho_k\|u_k\|_R^2+J^\star_{N-(n+1)}\left(Ax_n+\rho_nBu_n+\omega_n,\Theta_{n+1}\right)\,\middle|\,\Theta_0,x_0\right\}.$$

At stage N, we have

$$J_0^\star(x_N,\Theta_N)=\mathbf{E}\left\{x_N^TS_Nx_N|\Theta_N,x_N\right\}=x_N^TS_{N,i}x_N.$$

Now, at stage $k+1$, we have a cost of the form

$$J^\star_{N-(k+1)}(x_{k+1},\Theta_{k+1})=x_{k+1}^TS_{k+1,\Theta_{k+1}}x_{k+1}+c_{k+1,\Theta_{k+1}},$$

where $c_{k,\Theta_k}\triangleq\mathbf{E}\{c_k|\Theta_k\}$. Then, at stage k, we have

$$J^\star_{N-k}(x_k,\Theta_k)=\min_{u_k}\mathbf{E}\left\{\|x_k\|_Q^2+\rho_k\|u_k\|_R^2+J^\star_{N-(k+1)}(Ax_k+\rho_kBu_k+\omega_k,\Theta_{k+1})\,\middle|\,\Theta_k,x_k\right\}.$$

Writing this out yields

$$\begin{aligned}J^\star_{N-k}(x_k,\Theta_k)=\min_{u_k}\mathbf{E}\Big\{&x_k^T\left(Q+A^TS_{k+1,\Theta_{k+1}}A\right)x_k+\rho_ku_k^T\left(R+B^TS_{k+1,\Theta_{k+1}}B\right)u_k\\&+2\rho_ku_k^TB^TS_{k+1,\Theta_{k+1}}Ax_k+\omega_k^TS_{k+1,\Theta_{k+1}}\omega_k+c_{k+1,\Theta_{k+1}}\,\Big|\,\Theta_k,x_k\Big\},\end{aligned}$$

where we used the fact that $\rho_k^2=\rho_k$ and that ω_k is white. Computing the expectation yields

$$\begin{aligned}J^\star_{N-k}(x_k,\Theta_k)=\min_{u_k}\sum_{j\in\Xi}\Big(&x_k^T(Q+A^TS_{k+1,j}A)x_k+\mathbb{1}_{(1,\cdots,r)}(j)u_k^T(R+B^TS_{k+1,j}B)u_k\\&+2\mathbb{1}_{(1,\cdots,r)}(j)u_k^TB^TS_{k+1,j}Ax_k+\mathrm{trace}(\Sigma_\omega S_{k+1,j})+c_{k+1,j}\Big)\mathbf{Pr}\{\Theta_{k+1}=j|\Theta_k\},\end{aligned}$$

since ρ_k is only nonzero when $\Theta_{k+1}\le r$, as shown in Equation 3. By Equations 4 and 5, we have that $\mathbf{Pr}\{\Theta_{k+1}=j|\Theta_k\}$ is only nonzero when $j=\phi_{\Theta_k}$ and $j=\bar{\phi}_{\Theta_k}$. Utilizing this, results in

$$\begin{aligned}J^\star_{N-k}(x_k,\Theta_k)=\min_{u_k}\sum_{j=\left\{\phi_{\Theta_k},\bar{\phi}_{\Theta_k}\right\}}\Big(&x_k^T(Q+A^TS_{k+1,j}A)x_k+\mathrm{trace}(\Sigma_\omega S_{k+1,j})+c_{k+1,j}\\&+\mathbb{1}_{(1,\cdots,r)}(j)u_k^T(R+B^TS_{k+1,j}B)u_k+2\mathbb{1}_{(1,\cdots,r)}(j)u_k^TB^TS_{k+1,j}Ax_k\Big)\mathbf{Pr}\{\Theta_{k+1}=j|\Theta_k\}.\end{aligned}$$

By defining $\bar{S}_{k+1,i} \triangleq \sum_{j=\{\phi_i,\bar{\phi}_i\}} S_{k+1,j} p_{i,j}$, the above can be written as

$$J^{\star}_{N-k}(x_k,\Theta_k) = \min_{u_k} x_k^T (Q + A^T \bar{S}_{k+1,\Theta_k} A) x_k + \mathbf{Pr}\{\Theta_{k+1} = \phi_{\Theta_k} | \Theta_k\} \left(u_k^T \left(R + B^T S_{k+1,\phi_{\Theta_k}} B \right) u_k + 2 u_k^T B^T S_{k+1,\phi_{\Theta_k}} A x_k \right)$$
$$+ \operatorname{trace}(\Sigma_\omega \bar{S}_{k+1,\Theta_k}) + \sum_{j=\left\{\phi_{\Theta_k},\bar{\phi}_{\Theta_k}\right\}} c_{k+1,j} \mathbf{Pr}\{\Theta_{k+1} = j | \Theta_k\}. \tag{A1}$$

The u_k that minimizes Equation A1 is found by taking the gradient of Equation A1 with respect to u_k and setting it equal to 0. This yields

$$0 = 2\mathbf{Pr}\{\Theta_{k+1} = \phi_{\Theta_k} | \Theta_k\} \left(R + B^T S_{k+1,\phi_{\Theta_k}} B \right) u_k + 2\mathbf{Pr}\{\Theta_{k+1} = \phi_{\Theta_k} | \Theta_k\} B^T S_{k+1,\phi_{\Theta_k}} A x_k.$$

Here, the terms $\mathbf{Pr}\{\Theta_{k+1} = \phi_{\Theta_k} | \Theta_k\}$ cancel out. Isolating u_k results in Equations 8 and 9.

Substituting the optimal control (8) into Equation A1 results in

$$J^{\star}_{N-k}(x_k,\Theta_k) = x_k^T S_{k,\Theta_k} x_k + c_{k,\Theta_k}$$

where S_{k,Θ_k} is as in Equation 10 and c_{k,Θ_k} is given in Equation 12.

APPENDIX B

PROOF OF THEOREM 1

This proof is organized as follows. We first show that cost (22) can be expressed as a sum of the costs given the history $\hat{\Theta}_k$ for all $\hat{\Theta}_k \in \hat{\Xi}$. We then show that each of these costs can be stated as a function of the cost at the current stage plus the cost at the future stages. Finally, we show by induction that the optimal control laws of the form (21) are given by Equation 25.

To simplify the presentation of the proof, let

$$V_k \triangleq \|x_k\|_Q^2 + \rho_k \|\hat{u}_{k,v}\|_R^2.$$

Then, cost (22) at stage $N-k$ depending on Θ_k can be stated as

$$\begin{aligned}
\bar{J}_{N-k}\left(x_k, \bar{U}_{k,N}\right) &= \mathbf{E}\left\{ \sum_{\ell=k}^{N-1} \|x_\ell\|_Q^2 + \rho_\ell \|\hat{u}_{\ell,v}\|_R^2 + \|x_N\|_{S_0}^2 \,\middle|\, x_k \right\} \\
&= \mathbf{E}\left\{ \sum_{\ell=k}^{N-1} V_\ell + \|x_N\|_{S_0}^2 \,\middle|\, x_k \right\} \\
&= \sum_{v\in\hat{\Xi}} \mathbf{E}\left\{ \sum_{\ell=k}^{N-1} V_\ell + \|x_N\|_{S_0}^2 \,\middle|\, x_k, \hat{\Theta}_k = v \right\} \mathbf{Pr}\left\{ \hat{\Theta}_k = v \right\} \\
&= \sum_{v\in\hat{\Xi}} \mathbf{E}\left\{ J_{N-k}\left(x_k, \bar{U}_{k,N}, \Theta_k\right) \,\middle|\, x_k, \hat{\Theta}_k = v \right\} \hat{\pi}_v,
\end{aligned} \tag{B1}$$

where $\hat{\pi}_v \triangleq \mathbf{Pr}\left\{\hat{\Theta}_k = v\right\} = \sum_{i\in\hat{\Xi}_v} \pi_i$. Define cost (B1) conditioned on $\hat{\Theta}_k = v$ by

$$F_{N-k}\left(x_k, \bar{U}_{k,N}, v\right) \triangleq \mathbf{E}\left\{ J_{N-k}\left(x_k, \bar{U}_{k,N}, \Theta_k\right) \,\middle|\, x_k, \hat{\Theta}_k = v \right\} \hat{\pi}_v. \tag{B2}$$

This allows us to state (B1) as

$$\bar{J}_{N-k}\left(x_k, \bar{U}_{k,N}\right) = \sum_{v\in\hat{\Xi}} F_{N-k}\left(x_k, \left(\hat{u}_{k,v}, \bar{U}_{k+1,N}\right), v\right). \tag{B3}$$

In the next lemma, we show that the partial cost (B2) can be expressed as a function of the cost at stage $N-k$ plus the cost at stage $N-(k+1)$.

Lemma 1. *With control law (21), the partial costs (B2) can at each stage $N-k$ be expressed as*

$$F_{N-k}\left(x_k, \left(\hat{u}_{k,v}, \bar{u}_{k+1,N}\right), v\right) = \mathbf{E}\left\{ \mathcal{L}_{N-k}\left(x_k, \left(\hat{u}_{k,v}, \bar{u}_{k+1,N}\right), \Theta_k\right) \,\middle|\, x_k, \hat{\Theta}_k = v \right\} \hat{\pi}_v, \tag{B4}$$

where

$$\mathcal{L}_{N-k}\left(x_k,\left(\hat{u}_{k,v},\bar{U}_{k+1,N}\right),\Theta_k\right)\triangleq x_k^TQx_k+\rho_k\hat{u}_{k,v}^TR\hat{u}_{k,v}+J_{N-k-1}\left(Ax_k+\rho_kB\hat{u}_k+\omega_k,\bar{U}_{k+1,N},\Theta_{k+1}\right) \tag{B5}$$

and J_{N-k-1} is given in Equation 7.

Proof. For each $F_{N-k}\left(x_k,\left(\hat{u}_{k,v},\bar{U}_{k+1,N}\right),v\right)$, we have

$$\begin{aligned}F_{N-k}\left(x_k,(\hat{u}_{k,v},\bar{U}_{k+1,N}),v\right)&=\mathbf{E}\left\{\sum_{\ell=k}^{N-1}V_\ell+\|x_N\|_{S_0}^2\middle|x_k,\hat{\Theta}_k=v\right\}\hat{\pi}_v\\&=\mathbf{E}\left\{V_k\middle|x_k,\hat{\Theta}_k=v\right\}\hat{\pi}_v+\underbrace{\mathbf{E}\left\{\sum_{\ell=k+1}^{N-1}V_\ell+\|x_N\|_{S_0}^2\middle|x_k,\hat{\Theta}_k=v\right\}\hat{\pi}_v}_{C_k},\end{aligned} \tag{B6}$$

where computing the second expectation on the right-hand side yields

$$\begin{aligned}C_k&=\mathbf{E}\left\{\mathbf{E}\left\{\sum_{\ell=k+1}^{N-1}V_\ell+\|x_N\|_{S_0}^2\middle|x_{k+1},\Theta_{k+1}\right\}\middle|x_k,\hat{\Theta}_k=v\right\}\\&=\int\mathbf{E}\left\{\sum_{\ell=k+1}^{N-1}V_\ell+\|x_N\|_{S_0}^2\middle|x_{k+1},\Theta_{k+1}\right\}\mathbf{Pr}\left\{x_{k+1},\Theta_{k+1}\middle|x_k,\hat{\Theta}_k=v\right\}dx_{k+1}d\Theta_{k+1}\\&=\int J_{N-k-1}\left(x_{k+1},\bar{U}_{k+1,N},\Theta_{k+1}\right)\mathbf{Pr}\left\{x_{k+1},\Theta_{k+1}\middle|x_k,\hat{\Theta}_k=v\right\}dx_{k+1}d\Theta_{k+1}\\&=\mathbf{E}\left\{J_{N-k-1}\left(x_{k+1},\bar{U}_{k+1,N},\Theta_{k+1}\right)\middle|x_k,\hat{\Theta}_k=v\right\}.\end{aligned} \tag{B7}$$

The result then follows by substituting Equation B7 back into Equation B6. □

Now, we can proof Theorem 1 using an induction argument. At stage 0, we have the cost

$$\bar{J}_0^\star(x_N)=\mathbf{E}\left\{x_N^TS_Nx_N|x_N\right\}=\sum_{i\in\Xi}\bar{J}_{0,i}(x_N)\pi_i,$$

with $\bar{J}_{0,i}(x_N)$ defined in Equation B1. At stage $N-k-1$, we have with control laws of form (19) the optimal cost

$$\bar{J}_{N-k-1}^\star(x_{k+1})=\sum_{i\in\Xi}\bar{J}_{N-k-1,i}^\star(x_{k+1})\pi_i,$$

where $\bar{J}_{N-k-1,i}^\star(x_{k+1})\triangleq\bar{J}_{N-k-1,i}(x_{k+1},\bar{U}_{k+1,N}^\star)$ is given by

$$\bar{J}_{N-k-1,i}^\star(x_{k+1})=x_{k+1}^TS_{k+1,i}x_{k+1}+c_{k+1,i}.$$

Then, by Lemma 1, we have at stage $N-k$ for $\hat{\Theta}_k=v$ the cost

$$\begin{aligned}F_{N-k}^\star(x_k,v)&=\min_{\hat{u}_{k,v}}\mathbf{E}\left\{\mathcal{L}_{N-k,\Theta_k}(x_k,\hat{u}_{k,v})\middle|x_k,\hat{\Theta}_k=v\right\}\hat{\pi}_v\\&=\min_{\hat{u}_{k,v}}\sum_{i\in\Xi}\mathbf{E}\left\{\mathcal{L}_{N-k,\Theta_k(x_k},\hat{u}_{k,v})\middle|x_k,\Theta_k=i\right\}\mathbf{Pr}\{\Theta_k=i|\hat{\Theta}_k=v\}\hat{\pi}_v\\&\overset{(a)}{=}\min_{\hat{u}_{k,v}}\sum_{i\in\hat{\Xi}_v}\underbrace{\mathbf{E}\left\{\mathcal{L}_{N-k,\Theta_k}\left(x_k,\hat{u}_{k,v}\right)\middle|x_k,\Theta_k=i\right\}}_{D_i}\pi_i,\end{aligned} \tag{B8}$$

where in (a), we used Bayes' law. Now

$$\begin{aligned}D_i&=\mathbf{E}\left\{\mathbf{E}\left\{\mathcal{L}_{N-k,\Theta_k}\left(x_k,\hat{u}_{k,v}\right)\middle|x_k,\Theta_{k+1},\Theta_k\right\}\middle|x_k,\Theta_k=i\right\}\\&=\sum_{j\in\Xi}\mathbf{E}\left\{\mathcal{L}_{N-k,i}\left(x_k,\hat{u}_{k,v}\right)|x_k,\Theta_{k+1}=j\right\}p_{ij}\\&\overset{(a)}{=}\sum_{j\in\{\bar{\phi}_i,\phi_i\}}\mathcal{M}_{N-k,j}\left(x_k,\hat{u}_{k,v}\right)p_{ij},\end{aligned} \tag{B9}$$

where (a) follows by Equations 4 and 5 and

$$\begin{aligned}\mathcal{M}_{N-k,j}\left(x_k,\hat{u}_{k,v}\right) &\triangleq \mathbf{E}\left\{\mathcal{L}_{N-k,i}\left(x_k,\hat{u}_{k,v}\right)|x_k,\Theta_{k+1}=j\right\}\\ &= x_k^T\left(Q+A^TS_{k+1,j}A\right)x_k+\text{trace}(\Sigma_\omega S_{k+1,j})+c_{k+1,j}\\ &\quad+\mathbb{1}_{(r+1,\cdots,d)}(j)\hat{u}_{k,v}^T\left(R+B^TS_{k+1,j}B\right)\hat{u}_{k,v}+2\mathbb{1}_{(r+1,\cdots,d)}(j)\hat{u}_{k,v}^TB^TS_{k+1,j}Ax_k.\end{aligned}$$

Substituting Equation B9 back into Equation B8 yields

$$F_{N-k}^\star(x_k,v)=\sum_{i\in\hat{\Xi}_v}\mathcal{M}_{N-k,\bar{\phi}_i}(x_k,0)\,p_{i\bar{\phi}_i}\pi_i+\min_{\hat{u}_{k,v}}\mathcal{M}_{N-k,\phi_i}\left(x_k,\hat{u}_{k,v}\right)p_{i\phi_i}\pi_i.$$

Writing out $\mathcal{M}_{N-k,\bar{\phi}_i}$ and $\mathcal{M}_{N-k,\phi_i}$ yields

$$\begin{aligned}F_{N-k}^\star(x_k,v)=&\sum_{i\in\hat{\Xi}_v}\sum_{j\in\{\bar{\phi}_i,\phi_i\}}\left[x_{N-1}^T\left(Q+A^TS_{k+1,j}A\right)x_{N-1}+\text{trace}(\Sigma_\omega S_{k,j})+c_{k+1,j}\right]p_{ij}\pi_i\\ &+\min_{\hat{u}_{k,v}}\left[\sum_{i\in\hat{\Xi}_v}\left(\hat{u}_{k,v}^T\left(R+B^TS_{k+1,\phi_i}B\right)\hat{u}_{k,v}+2\hat{u}_{k,v}^TB^TS_{k,\phi_i}Ax_k\right)p_{i\phi_i}\pi_i\right].\end{aligned}\tag{B10}$$

Here, the last part can, using Equation 27, be rewritten to

$$\min_{\hat{u}_{k,v}}\hat{u}_{k,v}^T\left(\bar{R}_v+B^T\bar{S}_{k+1,v}B\right)\hat{u}_{k,v}+2\hat{u}_k^TB^T\bar{S}_{k+1,v}Ax_k.$$

Taking the gradient of Equation B10 with respect to $\hat{u}_{k,v}$ and setting this equal to 0, then results in Equations 24 and 25. Inserting the result into D_i in Equation B9 and rewriting this yields

$$\begin{aligned}D_i&=x_k^TS_{k,i}x_k+\sum_{j\in\{\phi_i,\bar{\phi}_i\}}\left(\text{trace}(\Sigma_\omega S_{k,j})+c_{k+1,j}\right)p_{ij}\\ &=x_k^TS_{k,i}x_k+c_{k,i}=\bar{J}_{N-k,i}^\star(x_k),\end{aligned}\tag{B11}$$

where $S_{k,i}$ is given in Equation 26 and $c_{k,i}$ in Equation 30. Combining Equations B3, B8, and B11, then results in Equation 29.