Optimal Control Design of a Reparable Multistate system

Saket Choudhary * **Nachikethas A. Jagadeesan** **

* *University of Southern California, LA, CA 90089 USA (e-mail: skchoudh@ usc.edu).*
** *University of Southern California, LA, CA 90089 USA (e-mail: nanantha@ usc.edu).*

Abstract:

Keywords:

1. PROBLEM DESCRIPTION

Model of a two state reparable system:

$$\frac{dp_0}{dt} = -\lambda_0 p_0(t) + \int_0^1 \mu_1(x) p_1(x,t)dx + \int_0^1 u^*(x,t)dx \quad (1)$$

$$\frac{\partial p_1(x,t)}{\partial t} + \frac{\partial p_1(x,t)}{\partial x} = -\mu_1(x)p_1(x,t) - u^*(x,t) \quad (2)$$

where:

$p_0(t)$: Probability that the device is in good mode 0 at time t.

$p_1(x,t)$: Probability density (with respect to repair time x) that the failed device is in failure mode 1 at time t and has an elapsed repair time of x

$\mu_1(x)$: Time-dependent nonnegative repair rate when the device is in failure state and has an elapsed repair time of x.

Given Initial Conditions:

$$p_1(x,0) = 0 \quad (3)$$
$$p_0(0) = 1 \quad (4)$$

Given Boundary Conditions:

$$p_1(0,t) = \lambda_0 p_0(t) \quad (5)$$
$$p_1(1,t) = 0 \quad (6)$$

The function u^* is given by

$$u^*(x,t) = (0.3 + 0.1sin(x))b(t) \quad (7)$$

The function $b(t)$ represents the input. It is related to the cost constraint function $c(t)$ as given below.

$$b(t) + \int_0^1 \mu_1(x) f(x,t)dx - 0.3p_0^*(t) = c(t) \quad (8)$$
$$f(x,t) = 0.1\cos(\pi t)\sin^2(1-x) \quad (9)$$

Our objective is to find the input $b(t)$ such that the resulting distribution $p_0(t)$ is closest (as measured by the 2-norm) to the optimal distribution $p_0^*(t)$ given below.

$$p_0^*(t) = 0.85 + 0.05\cos(2\pi t) \quad (10)$$

2. METHODOLOGY

We make couple of substitutions, following the notation that z_i^j refers to the value of z evaluated at time point i and at position j. The repair time is divided into m subintervals, while the system running time is divided into n subintervals. For the purposes of numerical implementation, we chose m and n to be 20 and 400 respectively.

So

$$p_0(t_j) = v_j \quad 0 \le j \le n \quad (11)$$
$$p_1(x_i, t_j) = w_j^i \quad 0 \le i \le m, 0 \le j \le n \quad (12)$$
$$\mu_1(x_i) = \mu^i \quad 0 \le i \le m \quad (13)$$
$$\lambda = \lambda_0 \quad (14)$$
$$(15)$$

Using the new notation, the boundary conditions and initial conditions may be written as follows.

Initial conditions:

$$w_0^i = 0 \quad \forall 0 \le i \le m \quad (16)$$
$$v_0 = 1 \quad (17)$$

Boundary Conditions:

$$w_j^{20} = 0 \quad \forall 0 \le j \le n \quad (18)$$
$$w_j^0 = \lambda v_j \quad (19)$$

Also condensing,

$$I_j^* = u^*(x_i, t_j) = g^i b_j \quad (20)$$
$$\int_0^1 u^*(x, t_j)dx = \alpha b_j \quad (21)$$
$$where\ g^i = (0.3 + 0.1sin(x^i))$$

And,

$$b(t) + \int_0^1 \mu_1(x) f(x,t) dx - 0.3 p_0^*(t) = c(t)$$

$$b_j = c_j - f_j \quad (22)$$

$$where\ f_j = \int_0^1 \mu_1(x) f(x,t_j) dx - 0.3 p_0^*(t_j) \quad (23)$$

Discretizing (1)

$$\frac{v_{j+1} - v_j}{\tau} = -\lambda v_j + I_j + I_j^* \quad (24)$$

$$I_j = h[\frac{\mu^0 w_j^0}{2} + \sum_{k=1}^{19} \mu^k w_j^k + \frac{\mu^{20} w_j^{20}}{2}] \quad (25)$$

$$= h[\frac{\mu^0 w_j^0}{2} + \sum_{k=1}^{19} \mu^k w_j^k]$$

$$I_j^* = \alpha b_j \quad (26)$$

Discretizing (2)

$$\frac{w_{j+1}^i - w_j^i}{\tau} + \frac{w_j^{i+1} - w_j^{i-1}}{2h} = -\mu^i w_j^i - g^i b_j$$

$$w_{j+1}^i = w_j^i - \frac{\tau}{2h}(w_j^{i+1} - w_j^{i-1}) - \tau\mu^i w_j^i - \tau g^i b_j \quad (27)$$

Applying LAX scheme $w_j^i = \frac{w_j^{i-1} + w_j^{i+1}}{2}$ we get,

$$\begin{aligned} w_{j+1}^i =& \left(\frac{w_j^{i+1} + w_j^{i-1}}{2}\right) - \frac{\tau}{2h}(w_j^{i+1} - w_j^{i-1}) \\ & - \tau\mu^i \left(\frac{w_j^{i+1} + w_j^{i-1}}{2}\right) \\ & - \tau g^i b_j \\ w_{j+1}^i =& \frac{1}{2}\left(1 - \mu^i\tau + \frac{\tau}{h}\right) w_j^{i-1} + \\ & \frac{1}{2}\left(1 - \mu^i\tau - \frac{\tau}{h}\right) w_j^{i+1} - \\ & \tau g^i b_j \end{aligned}$$

Under an appropriately defined matrix A, we can re-write the above equation to read

$$\boldsymbol{w}_{j+1} = A\boldsymbol{w}_j - b_j\tau\boldsymbol{g} + \boldsymbol{e_1} v_{j+1} \quad (28)$$

$$= (A)^{j+1}\boldsymbol{w}_0 - \left[\sum_{k=0}^{j} b_k (A)^{j-k}\right]\boldsymbol{g}\tau \quad (29)$$

$$+ \left[\sum_{k=0}^{j} v_{k+1}(A)^{j-k}\right]\boldsymbol{e_1}$$

where $\boldsymbol{e_1}$ is an $m \times 1$ matrix given by

$$\boldsymbol{e_1} = [\lambda, 0, \dots, 0]^T \quad (30)$$

Matrix A has the form:

$$\begin{pmatrix} w_{j+1}^0 \\ w_{j+1}^1 \\ \cdots \\ w_{j+1}^{n-2} \\ w_{j+1}^{n-1} \\ w_{j+1}^n \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ a_1 & 0 & a_3 & \cdots & 0 & 0 \\ 0 & a_2 & 0 & a_4 & \cdots & 0 \\ 0 & 0 & a_3 & 0 & a_5 & 0 \\ \vdots & & & & & \\ 0 & 0 & \cdots & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} w_j^0 \\ w_j^1 \\ \cdots \\ w_j^{n-2} \\ w_j^{n-1} \\ w_j^n \end{pmatrix} \quad (31)$$

$$+ b\tau \begin{pmatrix} g^0 \\ g^1 \\ \vdots \\ g^{n-2} \\ g^{n-1} \\ g^n \end{pmatrix} + v_{j+1} \begin{pmatrix} \lambda \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (32)$$

From (24)

$$v_{j+1} = (1 - \lambda\tau)v_j + \tau I_j + \tau I_j^* \quad (33)$$

$$= (1 - \lambda\tau + \frac{h\tau}{2})v_j + h\tau\boldsymbol{\mu}^T\boldsymbol{w}_j + \alpha b_j\tau \quad (34)$$

Substitute the expression for the time evolution for $\boldsymbol{w}$ in the above to obtain,

$$\begin{aligned} v_{j+1} =& (1 - \lambda\tau + \frac{h\tau}{2})v_j \\ & + h\tau\boldsymbol{\mu}^T(A)^j\boldsymbol{w}_0 \\ & - \boldsymbol{\mu}^T\left[\sum_{k=0}^{j-1} b_k(A)^{j-1-k}\right]\boldsymbol{g}\tau \\ & + \alpha b_j\tau \end{aligned}$$

Let's define

$$\beta_{j,k} = \boldsymbol{\mu}^T(A)^{j-1-k}\boldsymbol{g} \quad (35)$$

$$\omega_j = \boldsymbol{\mu}^T(A)^j\boldsymbol{w_0} \quad (36)$$

$$\gamma = (1 - \lambda\tau + \frac{h\tau}{2}) \quad (37)$$

$$\boldsymbol{\beta_0} = \begin{pmatrix} \alpha \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (38)$$

$$(39)$$

$$v_{j+1} = \gamma v_j + h\tau\omega_j - \tau\sum_{k=0}^{j-1}\beta_{j,k} b_k + \alpha b_j\tau \quad (40)$$

$$= \gamma v_j + h\tau\omega_j - \tau\boldsymbol{\beta}_j^T\boldsymbol{b} \quad (41)$$

$$= \gamma^{j+1}v_0 + h\tau\sum_{k=0}^{j}\gamma^{j-k}\omega_k - \left(\sum_{k=0}^{j}\gamma^{j-k}\boldsymbol{\beta}_k\right)^T\boldsymbol{b} \quad (42)$$

Under appropriately defined strictly lower triangular matrices G and B,

$$\boldsymbol{v} = -B\boldsymbol{b} + \boldsymbol{d} + h\tau G\boldsymbol{\omega} \tag{43}$$

$$\boldsymbol{d} = \begin{pmatrix} 1 \\ \gamma \\ \gamma^2 \\ \gamma^3 \\ \vdots \\ \gamma^n \end{pmatrix} v_0 \tag{44}$$

where row $j+1$ of matrix B is given by

$$\left(\sum_{k=0}^{j} \gamma^{j-k} \boldsymbol{\beta}_k \right)^T$$

From the initial conditions, we have $\boldsymbol{w_0} = \boldsymbol{0}$ Thus $\boldsymbol{\omega} = \boldsymbol{0}$. Consequently, we have,

$$\boldsymbol{v} = -B\boldsymbol{b} + \boldsymbol{d} \tag{45}$$

Note that $\boldsymbol{d}$ is a known quantity. Hence our optimization problem reduces to finding $\boldsymbol{b}$ that best fits the equation

$$B\boldsymbol{b} = \boldsymbol{d} - \boldsymbol{v}^* \tag{46}$$

The matrix B is invertible in this case and hence our estimate of $\boldsymbol{v}$ is given by

$$\hat{\boldsymbol{b}} = B^{-1} (\boldsymbol{d} - \boldsymbol{v}^*) \tag{47}$$

$$\boldsymbol{v} = -B\hat{\boldsymbol{b}} + \boldsymbol{d} \tag{48}$$

$$= \boldsymbol{v}^* \tag{49}$$

2.1 Computing the cost constraint function

We approximate the integral using:

$$\int_{-1}^{1} f(x,t)dx = \sum_{i=1}^{n} \rho_i f(x_i, t) \tag{50}$$

We used Gauss-Legendre quadrature with $n = 4$. Thus,

$$\int_{0}^{1} f(x,t)dx = \frac{1}{2} \sum_{i=1}^{n} \rho_i f\left(\frac{x_i}{2} + \frac{1}{2}, t \right) \tag{51}$$

$$\therefore c_j = b_j + f_j \tag{52}$$

3. OBSERVATION AND CONCLUSIONS

The inversion of matrix B does present some difficulties. It is easily seen from the definition of B that it is lower triangular. In addition,

$$(B)_{ii} = \alpha\tau \quad \forall 0 \le i \le n$$

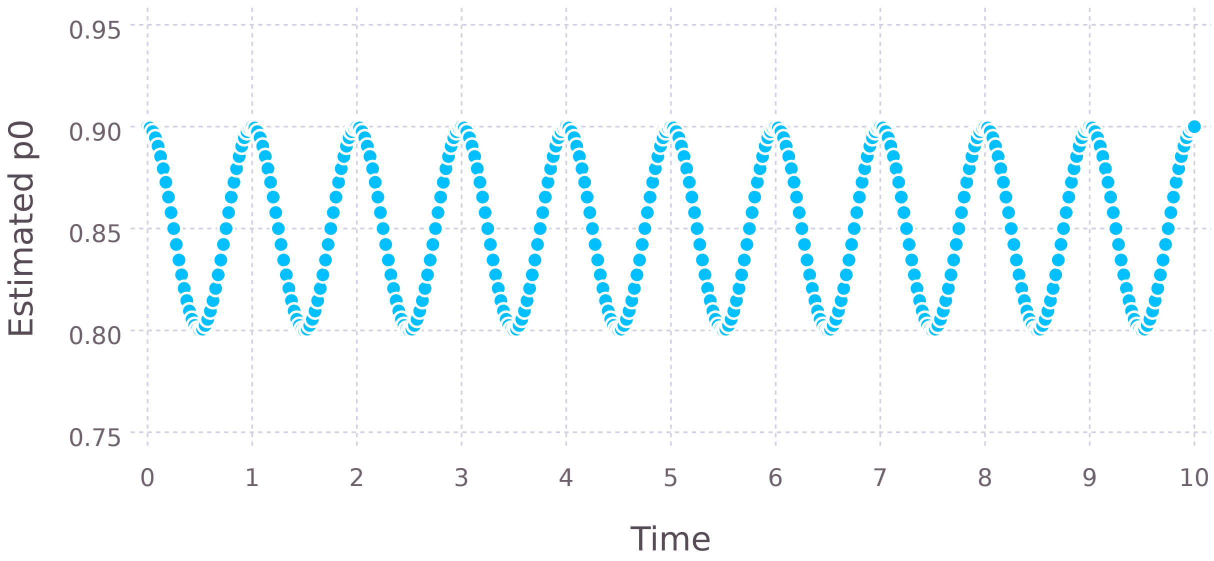
As B is lower triangular, its determinant is the product of the main diagonal terms.

$$|B| = (\alpha\tau)^{n+1}$$

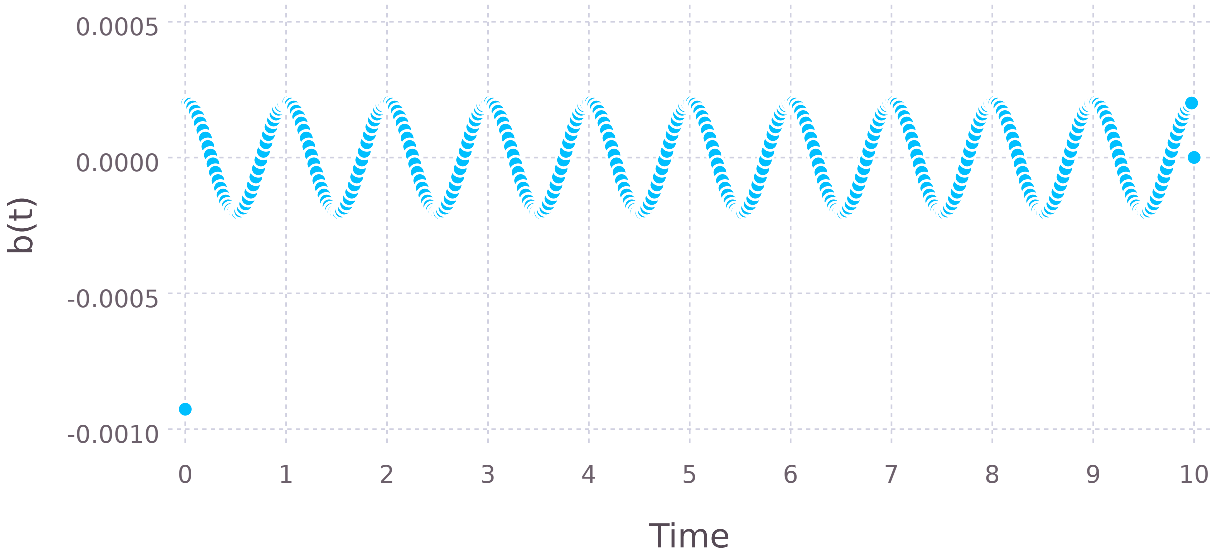
We have $\alpha \approx 0.354$, $\tau = 0.025$ and $n = 401$. Thus,

$$|B| \approx 10^{-802}$$

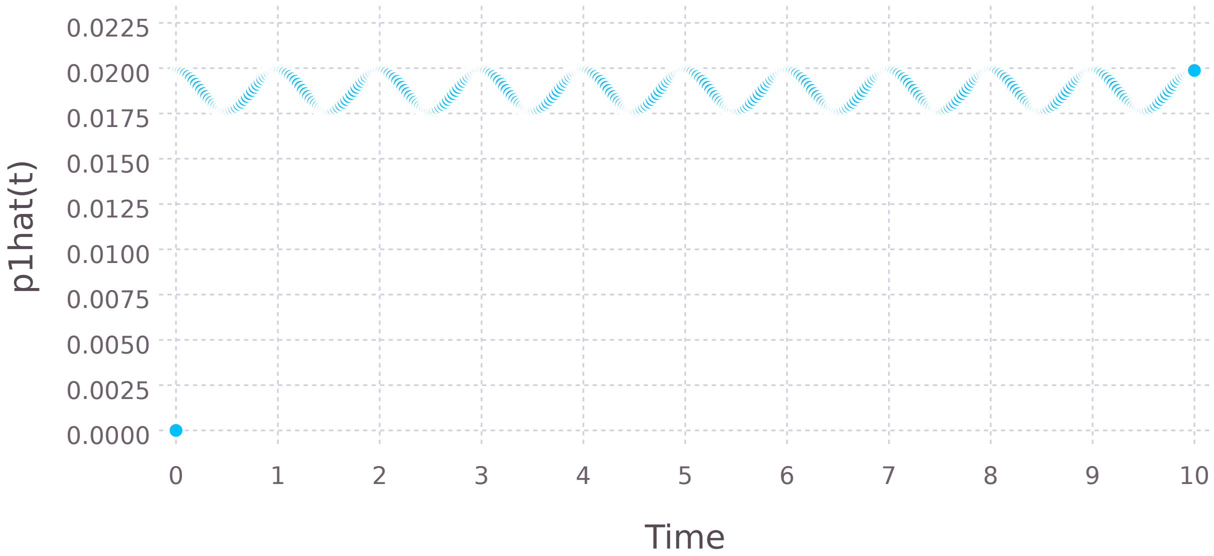
which makes B dangerously close to being singular. In fact, taking a direct inverse in Julia results in multiple entries going to `NaN`. We resolve this issue by computing the Moore-Penrose pseudoinverse B^+ of B. For an invertible matrix B, $B^- = B^+$

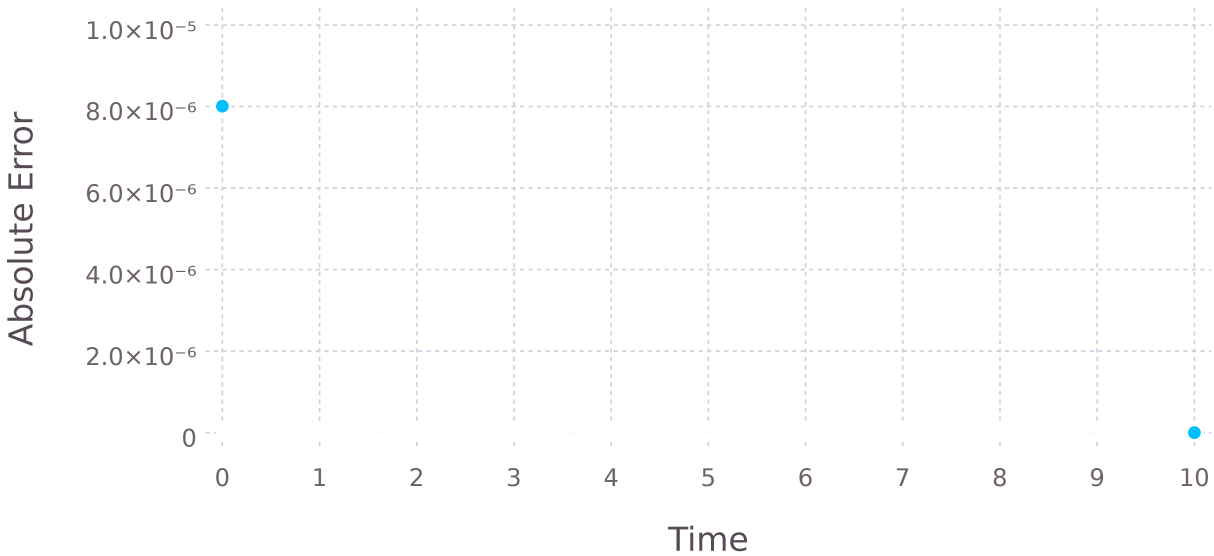


Estimated p0
0.95
0.90
0.85
0.80
0.75
0
1
2
3
4
5
6
7
8
9
10
Time



b(t)
0.0005
0.0000
-0.0005
-0.0010
0
1
2
3
4
5
6
7
8
9
10
Time


p1hat(t)
0.0225
0.0200
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
0 1 2 3 4 5 6 7 8 9 10
Time


Absolute Error
1.0×10⁻⁵
8.0×10⁻⁶
6.0×10⁻⁶
4.0×10⁻⁶
2.0×10⁻⁶
0
0 1 2 3 4 5 6 7 8 9 10
Time