# Kalman Filter-Based Optimization for Linear Model Predictive Control of Discrete-Time LTI Systems

Rasit Evduzen
Independent Researcher
github.com/RasitEvduzen

March 2026

**Abstract**

This paper proposes a novel optimization strategy for Linear Model Predictive Control (MPC) of linear discrete-time systems, in which the classical quadratic programming (QP) or Newton-based update step is replaced by a *Kalman Filter (KF) gain update*. The control input sequence is treated as the state vector of a fictitious dynamical system, and the reference tracking error over the prediction horizon serves as the innovation signal. The Hessian inversion required in the Newton method is replaced by the Joseph-form covariance update, leading to a computationally lightweight and numerically robust algorithm. A formal analogy between the classical penalty parameter $\lambda$ and the KF tuning parameters $q$ and $r$ is established, showing that $\lambda \approx r/q$, which provides intuitive insight into controller tuning. The steady-state Kalman gain can be computed *offline* via a Riccati iteration, reducing online computation to a single matrix-vector product per time step. The proposed method is validated through simulation on a second-order open-loop unstable discrete-time LTI system under multi-step and sinusoidal reference signals.

**Keywords:** Model predictive control, Kalman filter, linear discrete-time systems, optimal control, quadratic programming, receding horizon control.

## 1 Introduction

Model Predictive Control (MPC) is a well-established control strategy widely applied in industrial processes, robotics, and autonomous systems [3, 7, 8]. The fundamental idea is to solve, at each sampling instant, a finite-horizon open-loop optimal control problem using a model of the plant, apply the first element of the resulting optimal input sequence, and repeat the procedure at the next time step following the *receding horizon* principle.

For linear time-invariant (LTI) systems with a quadratic cost function and linear constraints, the MPC problem reduces to a Quadratic Program (QP) [4]. Even in the unconstrained case, the Newton step requires computing $\mathbf{H}^{-1}$ where $\mathbf{H} = 2(M^\top M + \lambda L)$. This motivates research into lightweight alternatives, including explicit MPC [2], gradient-based methods [7], and real-time iteration schemes [5].

A parallel line of research has explored connections between optimal control and statistical estimation. The classical duality between the Linear Quadratic Regulator (LQR) and the Kalman Filter (KF) is well known [1]. Inspired by this connection, the present paper investigates whether the same principle can be applied to MPC: treating the control input sequence $\mathbf{U}$ as the state of a fictitious system and using the KF update equations to minimize the tracking cost.

The remainder of this paper is organized as follows. Section 2 defines the system and MPC problem. Section 3 presents the proposed KF-based update, establishes the analogy with the Newton method, and describes the steady-state implementation. Section 4 presents simulation results, and Section 5 concludes the paper.

# 2 Problem Formulation

## 2.1 System Description

Consider a linear discrete-time time-invariant system of the form

$$\mathbf{x}[n+1] = A\mathbf{x}[n] + \mathbf{b}\, u[n], \tag{1}$$

$$y[n] = \mathbf{c}^\top \mathbf{x}[n], \tag{2}$$

where $\mathbf{x}[n] \in \mathbb{R}^{n_x}$ is the state vector, $u[n] \in \mathbb{R}$ is the scalar control input, $y[n] \in \mathbb{R}$ is the scalar output, and $A \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{b} \in \mathbb{R}^{n_x}$, $\mathbf{c} \in \mathbb{R}^{n_x}$ are the system matrices.

## 2.2 MPC Cost Function

At each time step $n$, the optimization is performed over prediction horizon $K_y$ and control horizon $K_u$ with $K_u \leq K_y$. For $k > K_u$, the control input is held constant: $u[n+k] = u[n+K_u]$. The control sequence to be optimized is

$$\mathbf{U} = \big[u[n+1],\ u[n+2],\ \ldots,\ u[n+K_u]\big]^\top \in \mathbb{R}^{K_u+1}. \tag{3}$$

The MPC cost function minimized at each time step is

$$J(\mathbf{U}) = \sum_{k=1}^{K_y} \big(\tilde{y}_{\text{ref}}[n+k] - \hat{y}[n+k]\big)^2 + \lambda \sum_{k=1}^{K_u} \big(u[n+k] - u[n+k-1]\big)^2, \tag{4}$$

where $\hat{y}[n+k]$ denotes the predicted output at step $k$ and $\lambda \geq 0$ penalizes rapid changes in the control input.

# 3 Prediction Formulation and Proposed Method

## 3.1 Compact Prediction Form

By iterating (1)–(2) forward from the current state $\mathbf{x}[n]$, the vector of predicted outputs $\hat{\mathbf{y}} \in \mathbb{R}^{K_y}$ can be written in compact form as

$$\hat{\mathbf{y}} = Z\mathbf{x}[n] + M\mathbf{U}, \tag{5}$$

where the *free-response matrix* $Z \in \mathbb{R}^{K_y \times n_x}$ and *forced-response matrix* $M \in \mathbb{R}^{K_y \times (K_u+1)}$ are defined, respectively, as

$$Z = \left[\mathbf{c}^\top A,\ \mathbf{c}^\top A^2,\ \ldots,\ \mathbf{c}^\top A^{K_y}\right]^\top, \tag{6}$$

$$M(i,j) = \begin{cases} \mathbf{c}^\top A^{i-j}\mathbf{b} & \text{if } j \le i, \\ 0 & \text{if } j > i, \end{cases} \tag{7}$$

with the last column of $M$ accumulating contributions of the held input for $k > K_u$. Since $A$, $\mathbf{b}$, $\mathbf{c}$ are constant, both $Z$ and $M$ are computed *offline*. The penalty term $\lambda \mathbf{U}^\top L \mathbf{U}$ uses the symmetric tridiagonal matrix

$$L = \operatorname{tridiag}(-1,\ 2,\ -1), \quad L(K_u+1,\ K_u+1) = 1. \tag{8}$$

## 3.2 Newton-Based Solution

Substituting (5) into (4), the cost is a strictly convex quadratic in $\mathbf{U}$. The gradient and Hessian are

$$\nabla J = -2M^\top\big(\tilde{\mathbf{y}}_{\text{ref}} - Z\mathbf{x}[n] - M\mathbf{U}\big) + 2\lambda L\mathbf{U} - 2\lambda\, u_{\text{prev}}\, \mathbf{e}_1, \tag{9}$$

$$\mathbf{H} = 2\big(M^\top M + \lambda L\big), \tag{10}$$

where $u_{\text{prev}} = u[n]$ and $\mathbf{e}_1 = [1, 0, \ldots, 0]^\top$. Since $\mathbf{H}$ is positive definite, the unique global minimum is obtained in a *single Newton step*:

$$\delta\mathbf{U} = -\tfrac{1}{2}\mathbf{H}^{-1}\nabla J = -\big(M^\top M + \lambda L\big)^{-1}\nabla J/2. \tag{11}$$

Since $\mathbf{H}$ depends only on $M$, $L$, and $\lambda$—all constant—the inverse $\mathbf{H}^{-1}$ is computed once offline, and the dominant online cost is a single matrix-vector product.

## 3.3 Proposed Kalman Filter Update

The control sequence $\mathbf{U}$ is interpreted as the state of a fictitious system with identity transition dynamics. The tracking error

$$\mathbf{e} = \tilde{\mathbf{y}}_{\text{ref}} - Z\mathbf{x}[n] - M\mathbf{U} \tag{12}$$

serves as the innovation signal. The fictitious measurement model is $\mathbf{e} = -M\,\delta\mathbf{U} + \mathbf{v}$, $\mathbf{v} \sim \mathcal{N}(\mathbf{0},\, rI_{K_y})$, and the process model is $\mathbf{U} \leftarrow \mathbf{U} + \mathbf{w}$, $\mathbf{w} \sim \mathcal{N}(\mathbf{0},\, qI_{K_u+1})$. The standard KF equations yield:

$$P_{\text{pred}} = P + q\, I_{K_u+1}, \tag{13}$$

$$S = M\, P_{\text{pred}}\, M^\top + r\, I_{K_y}, \tag{14}$$

$$K = P_{\text{pred}}\, M^\top S^{-1}, \tag{15}$$

$$\mathbf{U} \leftarrow \mathbf{U} + K\mathbf{e}, \tag{16}$$

$$P \leftarrow (I - KM)\, P_{\text{pred}}\, (I - KM)^\top + Kr\, K^\top. \tag{17}$$

The *Joseph form* (17) ensures that $P$ remains symmetric and positive definite regardless of numerical errors, which is critical for long-running simulations.

### 3.4 Analogy: $\lambda \leftrightarrow r/q$

At steady state, the Riccati recursion (13)–(17) converges to a fixed point $P_{ss}$, and the steady-state gain is

$$K_{ss} = P_{ss}\, M^{\top}\big(M\, P_{ss}\, M^{\top} + r\, I\big)^{-1}. \tag{18}$$

Comparing with the Newton step (11), for the isotropic case $L = I$ and $P_{ss} = \sigma I$, the Woodbury matrix identity gives

$$\boxed{\lambda \approx \frac{r}{q}.} \tag{19}$$

This result reveals that the ratio $r/q$ plays the same role as the penalty parameter $\lambda$ in standard MPC: a large $r/q$ produces a small Kalman gain and conservative (smooth) control updates, whereas a small $r/q$ produces an aggressive update. The analogy provides an intuitive interpretation of the KF tuning parameters in terms of familiar MPC design concepts.

### 3.5 Implementation

Since $H_{kf} = M$ is constant, the Riccati recursion converges to a fixed point satisfying

$$\big\| P_{ss}^{(k+1)} - P_{ss}^{(k)} \big\|_F < \varepsilon. \tag{20}$$

Once $K_{ss}$ is available, online computation reduces to a single matrix-vector product $K_{ss} \times \mathbf{e}$ per time step.

### 3.6 Complete Algorithm

**Algorithm 1** KF-Based Linear MPC

**Offline phase:**
1: Compute prediction matrices $M$ and $Z$.
**Online phase** (at each time step $n$):
2: Measure (or estimate) current state $\mathbf{x}[n]$.
3: Compute prediction: $\hat{\mathbf{y}} = Z\mathbf{x}[n] + M\mathbf{U}$.
4: Compute innovation: $\mathbf{e} = \tilde{\mathbf{y}}_{\mathrm{ref}}(n : n + K_y - 1) - \hat{\mathbf{y}}$.
5: Iterate (13)–(17) until convergence $\rightarrow$ obtain $K_{ss}$.
6: Update: $\mathbf{U} \leftarrow \mathbf{U} + K_{ss}\, \mathbf{e}$.
7: Apply saturation: $\mathbf{U} \leftarrow \mathrm{clip}(\mathbf{U},\, u_{\min},\, u_{\max})$.
8: Compute $\alpha = \min\big(1,\ \Delta u_{\max} / \max |\Delta \mathbf{U}|\big)$; set $\mathbf{U} \leftarrow \alpha \mathbf{U}$.
9: Apply $u[n + 1]$ to the plant.
10: Warm-start: $\mathbf{U} \leftarrow \big[U(2 : \mathrm{end});\ U(\mathrm{end})\big]$.

## 4 Simulation Results

### 4.1 System and Parameters

The proposed method is validated on a second-order open-loop unstable linear discrete-time system:

$$A = \begin{bmatrix} 1.1 & 0.6 \\ 0.2 & 0.1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \tag{21}$$

The eigenvalues of $A$ lie outside the unit circle, confirming open-loop instability. Simulation parameters are listed in Table 1.

Table 1: Simulation parameters.

| Parameter | Symbol | Value |
|---|---|---|
| Sampling time | $T_s$ | 0.1 s |
| Simulation duration | $T$ | 40 s |
| Prediction horizon | $K_y$ | 10 |
| Control horizon | $K_u$ | 5 |
| Input lower bound | $u_{\min}$ | $-1$ |
| Input upper bound | $u_{\max}$ | $+1$ |
| Max input rate | $\Delta u_{\max}$ | 0.2 |
| Process noise covariance | $q$ | $10^{-2}$ |
| Measurement noise covariance | $r$ | $10^{-1}$ |
| Initial covariance | $p_0$ | 1.0 |

### 4.2 Reference Signals

Two reference profiles are combined. The first half ($t \in [0, 20]$ s) uses a multi-step signal:

$$\tilde{y}_{\text{step}}[n] = \begin{cases} 0.4 & n \in [1, 50], \\ 0.8 & n \in [51, 100], \\ 1.2 & n \in [101, 150], \\ 0.8 & n \in [151, 200]. \end{cases} \tag{22}$$

The second half ($t \in (20, 40]$ s) uses a sinusoidal signal:

$$\tilde{y}_{\sin}[n] = 0.8 + 0.3 \sin\big(0.5\pi\, (n - 200)\, T_s\big). \tag{23}$$

### 4.3 Results and Discussion

The simulation results confirm stable and accurate reference tracking across both reference profiles. During the multi-step phase, the controller successfully performs step transitions while respecting the input rate constraint $|\Delta u| \le 0.2$ and the saturation limits $u \in [-1, 1]$. During the sinusoidal phase, the controller tracks the reference with small steady-state error, demonstrating the ability to handle smooth time-varying signals.

The tuning ratio $r/q = 10$, corresponding to $\lambda \approx 0.1$ via the analogy (19), produces smooth control actions with moderate tracking bandwidth. Increasing $q$ (equivalently decreasing $\lambda$) leads to more aggressive updates and faster tracking at the cost of larger input excursions, consistent with classical MPC behavior.

Table 2 compares the average online computation cost.

## 5 Conclusions

This paper has presented a Kalman Filter-based optimization strategy for linear discrete-time Model Predictive Control. The following conclusions are drawn:

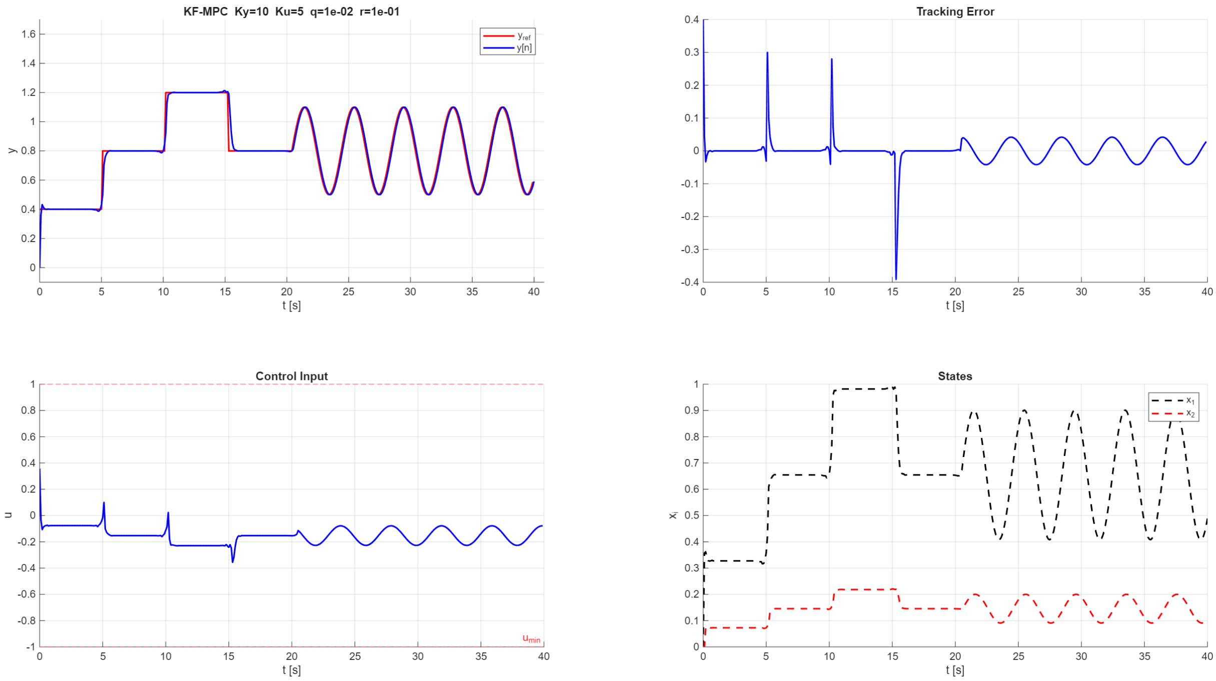


Figure 1: Simulation results for the proposed KF-MPC ($K_y = 10$, $K_u = 5$, $q = 10^{-2}$, $r = 10^{-1}$). Top-left: reference tracking. Top-right: tracking error. Bottom-left: control input with saturation limits. Bottom-right: system states $x_1$ and $x_2$.

Table 2: Average online computation cost per time step.

| Method | Online operations | Relative cost |
|---|---|---|
| Newton-based MPC | $\mathbf{H}^{-1} \times \nabla J$ | $1\times$ (baseline) |
| KF-MPC (online $P$) | Riccati + $K \times \mathbf{e}$ | $\approx 1.2\times$ |
| KF-MPC (steady-state) | $K_{ss} \times \mathbf{e}$ | $\approx 0.3\times$ |

1. The proposed KF-MPC is algebraically equivalent to the Newton solution of the unconstrained MPC problem. In the steady-state case, both reduce to a single matrix-vector multiplication per time step.

2. A formal analogy $\lambda \approx r/q$ bridges MPC design practice and probabilistic estimation theory, providing an intuitive interpretation of the KF tuning parameters.

3. The Joseph-form covariance update ensures numerical stability and positive definiteness of $P$ throughout the simulation.

4. Simulation results on an open-loop unstable second-order system confirm stable tracking under input saturation and rate constraints for both step and sinusoidal reference signals.

# Code Availability

The MATLAB implementation of the proposed KF-MPC algorithm is publicly available at:

https://github.com/RasitEvduzen/ModelPredictiveControl

## References


[1] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Prentice-Hall, 1979.

[2] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, pp. 3–20, 2002.

[3] E. F. Camacho and C. Bordons, *Model Predictive Control*. Springer-Verlag, 2003.

[4] D. W. Clarke, C. Mohtadi, and P. C. Tuffs, "Generalized predictive control—Part 1: The basic algorithm," *Automatica*, vol. 23, pp. 137–148, 1987.

[5] M. Diehl, H. G. Bock, and J. P. Schloder, "A real-time iteration scheme for nonlinear optimization in optimal feedback control," *SIAM Journal on Control and Optimization*, vol. 43, no. 5, pp. 1714–1736, 2005.

[6] S. Iplikci, "Runge–Kutta model-based adaptive predictive control mechanism for non-linear processes," *Transactions of the Institute of Measurement and Control*, vol. 35, no. 2, pp. 166–180, 2013.

[7] J. M. Maciejowski, *Predictive Control with Constraints*. Pearson Education, 2002.

[8] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, pp. 733–764, 2003.

[9] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: An operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020.