# A Sparse Implementation of an LQ Regulator using Back-Substitution

**Julia Adlercreutz**

**Abstract:** The Linear Quadratic Regulator is central in control theory, however, a weakness is that the input-output mat of the optimal controller almost always is dense, even when the problem data is sparse. In this work we reformulate LQR problem as a least squares problem. We demonstrate by example that while the input-output map is dense, the same control law can be computed using sparsity exploiting tools from linear algebra.

## 1. INTRODUCTION

The state-feedback solution to the Linear Quadratic Regulator (LQR) problem is a cornerstone of modern control theory (Kalman (1960)). In the discrete-time case it shows that under mild assumptions on the matrices $A, B$ and $\Pi$, the solution to the optimal control problem

$$\begin{aligned} \min_{u[0],u[1],u[2],\dots} & \sum_{k=0}^{\infty} \begin{bmatrix} x[k] \\ u[k] \end{bmatrix}^T \Pi \begin{bmatrix} x[k] \\ u[k] \end{bmatrix} \\ \text{s.t. } & x[k+1] = Ax[k] + Bu[k],\ x[0] \in \mathbb{R}^n \end{aligned} \tag{1}$$

is given by the static control law

$$u[k] = Kx[k]. \tag{2}$$

A difficulty in applying LQR in large-scale applications is that the optimal control law is almost always dense, even when the problem data (the matrices $A, B$ and $\Pi$) is not. This means that computing the $j$-th element of the control signal $u[k] \in \mathbb{R}^m$ requires the entire state measurement $x[k] \in \mathbb{R}^n$, which becomes infeasible if its dimension is too large. There has been significant developments in the area of exploiting sparsity in the problem data, for example through the concept of Quadratic Invariance Rotkowitz and Lall (2006); Lessard and Lall (2016), or more recently System Level Synthesis Wang et al. (2018) have been made.

In this paper, we take a slightly different perspective. Rather than trying to solve (1) under sparsity constraints on the control law, we instead look for sparse implementations of the standard LQR control law. This not only sidesteps all the theoretical issues and performance losses incurred by imposing sparsity constraints, but also preserves all the desirable robustness properties of the standard LQR solution.

The basic idea behind our approach is to rewrite (1) as a least squares problem of the form

$$\min_{v[0],v[1],\dots} \sum_{k=0}^{\infty} \|\mathfrak{q}^* \mathbf{M} v[k] + y_{\text{init}}[k]\|^2.$$

In the above the sequence $y$ and the operator $\mathbf{M}$ depend only on the problem data. Therefore we can obtain the optimal input from the closed form solution to the least squares problem. The hope is that the sparsity structure in the problem data is inherited by the operator $\mathbf{M}$. It is the objective of this note to present a simple example that shows that this approach is possible.

[1] The author is a member of the ELLIIT Strategic Research Area at Lund University and Linköping University. This work was supported by the ELLIIT Strategic Research Area.

## 2. RESULTS

### *2.1 The Example*

The rest of this note is focused on the following special instance of (1), in which $x \in \mathbb{R}^7$ and $u \in \mathbb{R}^3$.

$$\begin{aligned} \min_{u[0],u[1],\dots} & \sum_{k=0}^{\infty} r^{2k} \left( x_1[k]^2 + x_3[k]^2 + x_5[k]^2 + x_7[k]^2 \right) \\ \text{s.t.} \quad & x_{2i}[k+1] = u_i[k], \\ & x_{2i+1}[k+1] = x_{2i+1}[k] + x_{2i+2}[k] - u_i[k]. \end{aligned} \tag{3}$$

In the above the $0 < r < 1$ is a discount factor, and the index $i$ runs from 0 to 3 (where in a slight abuse, if the index is out of bounds, the term is ignored).

The underlying dynamics are sketched in Figure 1. The idea is that we have an underlying flow of a particular quantity (water, heat, ...) that we wish to regulate throughout a network by adjusting the amount of the quantity that is transported between neighbouring locations. This transportation is subject to delay.

### *2.2 A Controller Implementation using Back-Substitution*

In this section we will demonstrate that the optimal control law for (2) can be written as

$$K_1 u[0] = K_2 x[0],$$

where $K_1$ and $K_2$ are matrices that are banded in such a way that in order to compute the $i$-th input, only the state-variables in the neighbouring nodes and knowledge of the control input immediately upstream are required. It should be noted that $K_1^{-1} K_2 = K$.

In the rest of this section we will discover the reason for this structure in this control law. In particular we will see that the dynamic constraint can be given a sparse operator description. By extending some basic tools from numerical linear algebra to this setting, this allows the optimal control law to efficiently described and computed using back substitution.

**Least squares Problem Formulation:** The first step in deriving this control law is to reformulate (3) as a least squares problem. To do this we do a variable transform in the dynamics in (3), including the discount factor in the

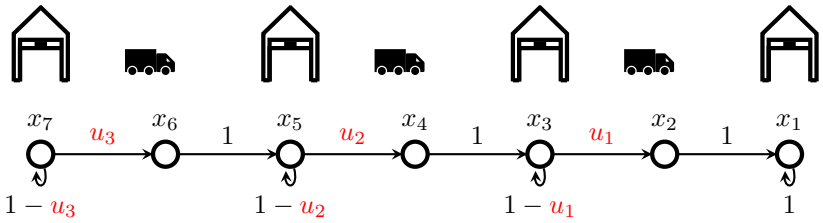


Fig. 1. Illustration of the system dynamics.

states. After that we reformulate the dynamics in terms of the backward shift operator $\mathfrak{q}^*$ so that the dynamics become

$$(1-r\mathfrak{q}^*)\begin{bmatrix}x_1[k]\\x_3[k]\\x_5[k]\\x_7[k]\end{bmatrix}=\mathfrak{q}^*\underbrace{\begin{bmatrix}r\mathfrak{q}^* & 0 & 0\\ -\mathbf{1} & r\mathfrak{q}^* & 0\\ 0 & -\mathbf{1} & r\mathfrak{q}^*\\ 0 & 0 & -\mathbf{1}\end{bmatrix}}_{\mathbf{M}}\begin{bmatrix}u_1[k]\\u_2[k]\\u_3[k]\end{bmatrix}+d[k],$$

where $d[k]=[Cx[0],\ CrAx[0],\ 0,\ 0,\ \ldots]$ captures the initial condition of the states. The operator matrix $\mathbf{M}$ captures the sparsity of the transportation network. The remaining dynamics are not important for what follows and is therefore hidden by the invertible variable transforms

$$\begin{cases}(\mathbf{1}-r\mathfrak{q}^*)\,v[k] & =u[k]\\ (\mathbf{1}-r\mathfrak{q}^*)\,y_{\text{init}}[k] & =d[k].\end{cases}$$

Notice that $v[0]=u[0]$, which is the control law we are looking for and the

$$y_{\text{init}}[k]=\left[Cx[0],\ CrAx[0],\ Cr^2Ax[0],\ Cr^3Ax[0],\ \ldots\right].$$

This means that the LQR problem in(3) can be rewritten to the least squares problem

$$\min_{v[0],v[1],\ldots}\sum_{k=0}^{\infty}\|\mathfrak{q}^*\mathbf{M}v[k]+y_{\text{init}}[k]\|^2.$$

The solution to this problem is

$$\mathbf{M}^*\mathbf{M}\hat{v}[k]=-\mathfrak{q}\mathbf{M}^*e[k],$$

where $\hat{v}[k]$ denotes the optimal solution and $\mathbf{M}^*$ denotes the adjoint of $\mathbf{M}$ which you get by transposing the matrix and replacing the backward shift operators with their adjoint operator, the forward shift $\mathfrak{q}$.

**Cholesky-factorisation:** In the previous subsection we saw that (3) could be formulated as a least squares problem involving a sparse operator $\mathbf{M}$. In this section we will show how to obtain a sparse Cholesky-factorisation of $\mathbf{M}^*\mathbf{M}$ into $\mathbf{L}\mathbf{L}^*$, where $\mathbf{L}$ is lower triangular.

To perform an Cholesky-factorisation, we need the operations of addition, subtraction, adjoints, multiplication, inversion and finding square roots. In general it can be hard to find inverses and square roots of operators built out of shift operators, but in this example it is not.

The Cholesky factorisation is given on the form

$$\mathbf{L}=\begin{bmatrix}l_{11}\mathbf{1} & \mathbf{0} & \mathbf{0}\\ l_{21}\mathfrak{q} & l_{22}\mathbf{1} & \mathbf{0}\\ \mathbf{0} & l_{32}\mathfrak{q} & l_{33}\mathbf{1}\end{bmatrix},$$

where $\mathbf{L}\mathbf{L}^*=\mathbf{M}^*\mathbf{M}$ and $l_{ij}\in\mathbb{R}$.

**Solving for $u$ using Back-substitution:** We can now exploit our sparse Choelsky-factorisation to compute the optimal control input using back-substitution. We are specifically interested in $\hat{u}[0]=\hat{v}[0]$.

First we compute

$$w[k]=-\mathfrak{q}\mathbf{M}^*y_{\text{init}}[k]$$

Because of the structure in $y_{\text{init}}$ it can be seen that $w[k]=r^kw[0]$. To find the expression for $w[0]$ we express $\mathbf{M}^*=(M^*)_0+(M^*)_1\mathfrak{q}$. Then it can be shown that

$$w[0]=\underbrace{\left((M^*)_0+(M^*)_1 r\right)CrA}_{K_2}x[0].$$

Next we need to solve $\mathbf{L}\mathbf{L}^*v=w$. As usual, we define an intermediate variable, and first solve $\mathbf{L}\nu=w$. Note that $\mathbf{L}^*$ does not conatin any forward shifts. This means that we only need $\nu[0]$ to compute $v[0]$.

To do this we express $\mathbf{L}=L_0+L_1\mathfrak{q}$ and reformulate the equation to solve into

$$\underbrace{\begin{bmatrix}L_0 & L_1 & 0\\ 0 & L_0 & L_1\\ 0 & 0 & L_0\end{bmatrix}}_{\mathbf{L}_T}\begin{bmatrix}\nu[0]\\ \overline{\nu}[1]\\ \overline{\nu}[2]\end{bmatrix}=\begin{bmatrix}w[0]\\ w[1]\\ w[2]\end{bmatrix},$$

where the notation $\overline{\nu}[k]$ indicates that the we do not get the correct signal for that sample. In order to get the entyer signal correct the matrix $\mathbf{L}_T$ would have to be semi-infinite, but because the matrix $L_0$ is diagonal and $L_1$ is lower triangular with zeros on the diagonal the first values can be obtained with a truncated $\mathbf{L}_T$.

Performing the back substitution we obtain that

$$v[0]=\underbrace{\left(L_0^{-1}\right)^2\left(I-\overline{L}r+\overline{L}^2r^2\right)}_{K_1^{-1}}K_2x[0],$$

where $\overline{L}=L_0^{-1}L_1$.

## REFERENCES


Kalman, R. (1960). Contributions to the theory of optimal control. *Boletin de la Sociedad Matématica Mexicana*, 102–119.

Lessard, L. and Lall, S. (2016). Convexity of decentralized controller synthesis. *IEEE Transactions on Automatic Control*, 61(10), 3122–3127. doi: 10.1109/TAC.2015.2504179.

Rotkowitz, M. and Lall, S. (2006). A characterization of convex problems in decentralized control. *IEEE Transactions on Automatic Control*, 51(2), 274–286. doi: 10.1109/TAC.2005.860365.

Wang, Y., Matni, N., and Doyle, J.C. (2018). Separable and localized system-level synthesis for large-scale systems. *IEEE Transactions on Automatic Control*, 63(12), 4234–4249. doi:10.1109/TAC.2018.2819246.