1 ZMP LQR Riccati Equation

Using $z(t)$ as the 2D position of the ZMP, we formulate:

$$
\begin{aligned}
\underset{u(t)}{\text{minimize}} \quad & \int_0^\infty \left[\|z(t) - z_d(t)\|_2^2 + \|u(t)\|_R^2\right] dt, \\
\text{subject to} \quad & R = R' > 0, \\
& z_d(t) = z_d(t_f), \quad \forall t \geq t_f \\
& \dot{x}(t) = Ax(t) + Bu(t), \quad z(t) = Cx(t) + Du(t) \\
& A = \begin{bmatrix} 0_{2\times2} & I_{2\times2} \\ 0_{2\times2} & 0_{2\times2} \end{bmatrix}, \quad B = \begin{bmatrix} 0_{2\times2} \\ I_{2\times2} \end{bmatrix} \\
& C = \begin{bmatrix} I_{2\times2} & 0_{2\times2} \end{bmatrix}, \quad D = -\frac{h}{g} I_{2\times2}
\end{aligned}
$$

This can be rewritten as a cost on state, *in coordinates relative to the final conditions*, $\bar{x} = x - \begin{bmatrix} z_d^T(t_f) & 0 & 0 \end{bmatrix}^T$, $\bar{z}_d(t) = z_d(t) - z_d(t_f)$:

$$
\begin{aligned}
\underset{u(t)}{\text{minimize}} \quad & \int_0^\infty \bar{x}^T Q_1 \bar{x} + \bar{x}^T q_2(t) + q_3(t) + u^T R_1 y + u^T r_2(t) + 2\bar{x}^T N u \\
\text{subject to} \quad & Q_1 = \text{diag}(1 \quad 1 \quad 0 \quad 0), \quad q_2(t) = \begin{bmatrix} -2\bar{z}_d(t) \\ 0_{2\times1} \end{bmatrix}, \quad q_3(t) = \|\bar{z}_d(t)\|_2^2 \\
& R_1 = R + \left(\frac{h}{g}\right)^2 I_{2\times2}, \quad r_2(t) = 2\bar{z}_d(t)\frac{h}{g}, \quad N = -\frac{h}{g}\begin{bmatrix} I_{2\times2} \\ 0_{2\times2} \end{bmatrix} \\
& \dot{x}(t) = Ax(t) + Bu(t) \\
& A = \begin{bmatrix} 0_{2\times2} & I_{2\times2} \\ 0_{2\times2} & 0_{2\times2} \end{bmatrix}, \quad B = \begin{bmatrix} 0_{2\times2} \\ I_{2\times2} \end{bmatrix}
\end{aligned}
$$

Note that this implies that $\bar{x}(\infty) = 0$ in order for the cost to be finite.

The resulting cost-to-go is given by

$$J = \bar{x}^T S_1(t)\bar{x} + \bar{x}^T s_2(t) + s_3(t),$$

with the corresponding Riccati differential equation given by

$$
\begin{aligned}
\dot{S}_1 &= -\left(Q_1 - (N + S_1 B)R_1^{-1}(B^T S_1 + N^T) + S_1 A + A^T S_1\right) \\
\dot{s}_2 &= -\left(q_2(t) - 2(N + S_1 B)R^{-1} r_s(t) + A^T s_2\right), \quad r_s(t) = \frac{1}{2}(r_2(t) + B^T s_2(t)) \\
\dot{s}_3 &= -\left(q_3(t) - r_s(t)^T R^{-1} r_s(t)\right)
\end{aligned}
$$

Note that S_1 has no time-dependent terms, and therefore $S_1(t)$ is a constant, given by the steady-state solution of the algebraic Riccati equation (e.g. from time-invariant LQR). Similarly, the feedback controller is given by

$$u(t) = K_1(t)\bar{x} + k_2(t),$$

and again the feedback $K_1(t)$ is a constant (derived from the infinite horizon LQR with Q, R, and N set as above).

1.1 Solving for $s_2(t)$

Given this, the affine terms in the Riccati differential equation are given by the linear differential equations:

$$\dot{s}_2(t) = A_2 s_2(t) + B_2 \bar{z}_d(t), \quad s_2(t_f) = 0$$

with

$$A_2 = (N + S_1 B)R^{-1}B^T - A^T, \quad B_2 = \begin{bmatrix} 2I_{2\times2} \\ 0_{2\times2} \end{bmatrix} + 2\frac{h}{g}(N + S_1 B)R^{-1}$$

Assuming $\bar{z}_d(t)$ is described by a *continuous* piecewise polynomial of degree k with $n+1$ breaks at t_j (with $t_0 = 0$ and $t_n = t_f$):

$$\bar{z}_d(t) = \sum_{i=0}^{k} c_{j,i}(t - t_j)^i, \quad \text{for } j = 0, ..., n-1, \text{ and } \forall t \in [t_j, t_{j+1}),$$

this system has a closed-form solution given by:

$$s_2(t) = e^{A_2(t-t_j)}\alpha_j + \sum_{i=0}^{k} \beta_{j,i}(t - t_j)^i, \quad \forall t \in [t_j, t_{j+1}),$$

with α_j and $\beta_{j.i}$ vector parameters to be solved for. Taking

$$\begin{aligned} \dot{s}_2(t) =& A_2 e^{A_2(t-t_j)}\alpha_j + \sum_{i=0}^{k} A_2\beta_{j,i}(t - t_j)^i + \sum_{i=0}^{k} B_2 c_{j,i}(t - t_j)^i \\ =& A_2 e^{A_2(t-t_j)}\alpha_j + \sum_{i=1}^{k} i\beta_{j,i}(t - t_j)^{i-1} \end{aligned}$$

forces that

$$\begin{gathered} A_2\beta_{j,i} + B_2 c_{j,i} = (i+1)\beta_{j,i+1}, \quad \text{for } i = 0, ..., k-1 \\ A_2\beta_{j,k} + B_2 c_{j,k} = 0. \end{gathered}$$

Note: need to prove that A_2 is full rank (it appears to be in practice). Solve backwards ($i = k, k-1, ..., 0$) for $\beta_{j,i}$. Finally, the continuity and the terminal boundary condition $s(t_f) = 0$ gives

$$e^{A(t_{j+1}-t_j)}\alpha_j + \sum_{i=0}^{k} \beta_{j,i}(t_{j+1} - t_j)^{i+1} = s(t_{j+1}).$$

1.2 Reading out $k_2(t)$

The remaining term for the controller is a simple read-out given the solution to $s_2(t)$:

$$k_2(t) = -\frac{h}{g} R^{-1} \bar{z}_d(t) - \frac{1}{2} R^{-1} B^T s_2(t)$$

which can be written as

$$k_2(t) = \alpha_L e^{A_2(t-t_j)} \alpha_{j,R} + \sum_{i=0}^{k} \gamma_{j,i} (t - t_j)^i$$

with

$$\alpha_L = -\frac{1}{2} R^{-1} B^T$$
$$\alpha_R = \alpha_{j,R} = \alpha_j$$
$$\gamma_{j,i} = -\frac{h}{g} R^{-1} c_{j,i} - \frac{1}{2} R^{-1} B^T \beta_{j,i}$$

1.3 Solving for $x_{com}(t)$

The resulting system is

$$\dot{x} = Ax + B\left(K_1 x + k_2(t)\right) = (A + BK_1)x + Bk_2(t),$$

where $x = [x_{com}, y_{com}, \dot{x}_{com}, \dot{y}_{com}]^T$. Since the solution $k_2(t)$ is the result of another linear system (cascaded in front of this one), it is easiest for me to solve jointly, using $y = \begin{bmatrix} x \\ s_2 \end{bmatrix}$:

$$\dot{y} = A_y y + B_y \bar{z}_d$$
$$A_y = \begin{bmatrix} A + BK_1 & -\frac{1}{2} B R^{-1} B^T \\ 0 & A_2 \end{bmatrix}, \quad B_y = \begin{bmatrix} -\frac{h}{g} B R^{-1} \\ B_2 \end{bmatrix}$$

$$y(t) = e^{A_y(t-t_j)} a_j + \sum_{i=0}^{k} b_{j,i} (t - t_j)^i$$

i can solve for b using the same technique as above (and re-using the β sol), and then solve for the top half of a_j forward in time.

1.4 Solving for $s_3(t)$

Having solved for $s_2(t)$, the dynamics of $s_3(t)$ in segment j can be written as

$$\dot{s}_3(t) = \bar{z}_d^T(t) \left(\frac{h^2}{g^2} R^{-1} - I \right) \bar{z}_d(t) + \frac{1}{4} s_2^T(t) B R^{-1} B^T s_2(t) + \frac{h}{g} \bar{z}_d^T(t) R^{-1} B^T s_2(t)$$

Let us rewrite our vector polynomials as, for instance, $\bar{z}_d(t) = \vec{c}_j m_k(t - t_j)$, with

$$\mathbf{c}_j = \begin{bmatrix} c_{j,0} & c_{j,1} & \ldots & c_{j,k} \end{bmatrix}$$

and

$$m_k(t) = \begin{bmatrix} 1 \\ t \\ t^1 \\ \ldots \\ t^k \end{bmatrix}.$$

We will also use the fact that

$$(e^{At})^T e^{At} = e^{(A^T + A)t}.$$

Then we have (dropping the j's for notational convenience and leaving $t_j = 0$):

$$\begin{aligned}
\dot{s}_3(t) =& m_k^T(t) \left[\mathbf{c}^T \left(\frac{h^2}{g^2} R^{-1} - I \right) \mathbf{c} + \frac{1}{4} \beta^T \beta \right] m_k(t) + \frac{1}{4} \alpha^T e^{A_2^T t} B^T R^{-1} B e^{A_2 t} \alpha \ldots \\
&+ m_k^T(t) \left[\frac{1}{2} \beta^T B + \frac{h}{g} \mathbf{c}^T \right] R^{-1} B^T e^{A_2 t} \alpha
\end{aligned}$$

The integral of this is ugly, and the requirement for accuracy here is less strict. For parsimony, we will approximate $s_3(t)$ with a Hermite cubic spline with the values and derivatives set (analytically) at the breakpoints of the desired ZMP trajectory. This means that we can cut a few computational corners in order to evaluate the value of $s_3(t)$ at the breakpoints, instead of maintaining the entire closed-form solution. Note that $\dot{s_3}(t)$ is continuous - the left and right derivates are equal.

We'll make use of the following steps to complete the integral:

$$\frac{d}{dt} m_k(t) = \begin{bmatrix} 0 \\ 1 \\ 2t \\ \vdots \\ kt^{k-1} \end{bmatrix} = \begin{bmatrix} 0_{1\times k} \\ \mathrm{diag}(1, 2, \cdots, k) \end{bmatrix} m_{k-1}(t) \equiv D_k m_{k-1}(t)$$

$$\int m_k(t) dt = \begin{bmatrix} t \\ \frac{1}{2} t^2 \\ \vdots \\ \frac{1}{k+1} t^{k+1} \end{bmatrix} = \begin{bmatrix} 0_{k+1\times 1} & \mathrm{diag}(1, \frac{1}{2}, \cdots, \frac{1}{k+1}) \end{bmatrix} m_{k+1}(t) \equiv D_{k+1}^{\sharp} m_{k+1}(t)$$

$$\begin{aligned}
\int_0^a m_k(t) dt =& D_{k+1}^{\sharp} m_{k+1}(a) \\
&\text{Note :} D_k \text{ is } k+1 \times k, D_k^{\sharp} \text{ is } \mathrm{pinv}(D_k) \\
\int e^{At} dt =& A^{-1} e^{At}
\end{aligned}$$

$$\int_0^a m_k^T(t) P m_k(t) dt = \int_0^a \mathrm{Tr}(P m_k(t) m_k^T(t)) dt = \int_0^a \mathrm{vec}(P^T)^T \mathrm{vec}(m_k(t) m_k^T(t)) dt$$