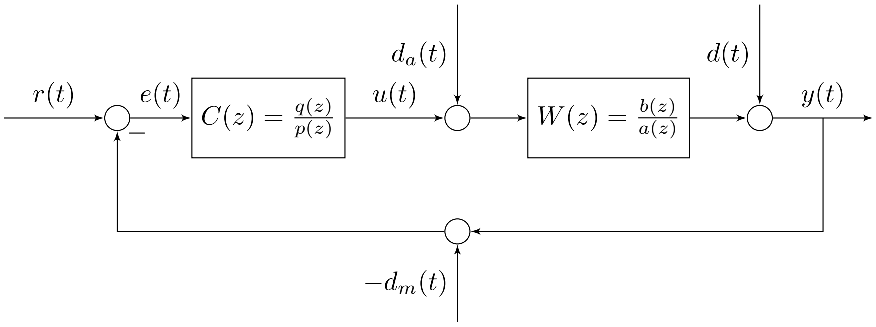
Polynomial approach to closed-loop control

F. A. Ramponi
Rev. 0.0.6, 2019-11-06

0.1 Simple closed loop controller

Our goals in polynomial closed-loop control (as opposed, say, to the standard state-space approach, decoupled state feedback and Luenberger observer) are manifold: first, to stabilize the plant (and the closed loop); second, to attain reference tracking; and third, to reject disturbances. Here is a complete control scheme:



In the diagram,

- $W(z)$ is the transfer function of the plant; we assume that it is *rational* and, as is usual in control design, also *strictly proper*, so that there is at least a delay in the closed loop:

$$W(z) = \frac{b(z)}{a(z)}, \qquad a, b \text{ polynomials}, \qquad \deg(b) < \deg(a);$$

- $C(z)$ is the transfer function of the controller (i.e. of what is going to become the *control unit* of the self-tuning regulator); we assume that it is rational and proper, but not necessarily in the strict sense;

$$C(z) = \frac{q(z)}{p(z)}, \qquad p, q \text{ polynomials}, \qquad \deg(q) \leq \deg(p);$$

- $r(t)$ is the reference signal;
- $u(t)$ is the control input;
- $y(t)$ is the output of the plant;
- $d_a(t)$ is a disturbance acting on the control actuator;
- $d(t)$ is a disturbance acting on the output (e.g. a load);
- $d_m(t)$ is a measurement error, corrupting the comparison between $y(t)$ and $r(t)$ that is the controller's main focus; note that the term $-d_m(t)$ is added with a minus sign only for ease of notation, so that the two '$-$' cancel out and the disturbance actually enters the loop with a positive sign;
- $e(t)$ is the tracking error.

Later in the section, for simplicity, we will ignore the role of $d_a(t)$ and $d_m(t)$ and get to a simpler scheme containing only the disturbance $d(t)$; but it must be clear from the beginning that all the disturbances are *always* present in real-world control loops. And hence, to prevent disturbances from de-stabilizing the closed-loop system, we must impose the BIBO stability not just of the "standard" closed-loop transfer function, but of four distinct transfer functions:

$$W_{\text{cl}}(z) = W_{r\to y}(z) = \frac{C(z)W(z)}{1+C(z)W(z)} = \frac{\frac{b(z)}{a(z)}\frac{q(z)}{p(z)}}{1+\frac{b(z)}{a(z)}\frac{q(z)}{p(z)}} = \frac{b(z)q(z)}{a(z)p(z)+b(z)q(z)} \qquad \text{(closed-loop transfer function);}$$

$$S(z) = W_{d\to y}(z) = \frac{1}{1+C(z)W(z)} = \frac{1}{1+\frac{b(z)}{a(z)}\frac{q(z)}{p(z)}} = \frac{a(z)p(z)}{a(z)p(z)+b(z)q(z)} \qquad \text{(sensitivity function);}$$

$$S_m(z) = W_{r\to u}(z) = \frac{C(z)}{1+C(z)W(z)} = \frac{\frac{q(z)}{p(z)}}{1+\frac{b(z)}{a(z)}\frac{q(z)}{p(z)}} = \frac{a(z)q(z)}{a(z)p(z)+b(z)q(z)} \qquad \text{(``measurement sensitivity'');}$$

$$S_a(z) = W_{u\to y}(z) = \frac{W(z)}{1+C(z)W(z)} = \frac{\frac{b(z)}{a(z)}}{1+\frac{b(z)}{a(z)}\frac{q(z)}{p(z)}} = \frac{b(z)p(z)}{a(z)p(z)+b(z)q(z)} \qquad \text{(``actuation sensitivity'');}$$

Note that $W_{\text{cl}}(z)$ and $S(z)$ satisfy the constraint $W_{\text{cl}}(z) + S(z) = 1$.

I have given $S_m(z)$ and $S_a(z)$ the silly names "measurement sensitivity" and "actuation sensitivity" because

$$S_m(z) = W_{r\to u}(z) \equiv W_{d_m\to u}(z) \quad \text{and}$$
$$S_a(z) = W_{u\to y}(z) \equiv W_{d_a\to y}(z),$$

as you are invited to check. The name "sensitivity function" is instead classical, and has a technical reason related to the first studies about *robustness*. Suppose that the transfer function W of the plant is "perturbed" with a small variation ΔW; then $\Delta W/W$ is its *relative* perturbation. How does this variation affect the closed-loop transfer function? Let ΔW_{cl} be the variation in W_{cl} corresponding to ΔW. Treating W_{cl} and W symbolically (or considering them at a fixed complex frequency) we get:

$$\begin{aligned}
\frac{\Delta W_{\text{cl}}/W_{\text{cl}}}{\Delta W/W} &= \frac{\Delta W_{\text{cl}}}{\Delta W}\cdot\frac{W}{W_{\text{cl}}} \simeq \frac{dW_{\text{cl}}}{dW}\cdot\frac{W}{W_{\text{cl}}} \\
&= \left(\frac{d}{dW}\frac{CW}{1+CW}\right)\cdot\left(\frac{W(1+CW)}{CW}\right) = \frac{C}{(1+CW)^2}\cdot\frac{(1+CW)}{C} = \frac{1}{(1+CW)} = S; \\
\frac{\Delta W_{\text{cl}}}{W_{\text{cl}}} &\simeq S\cdot\frac{\Delta W}{W},
\end{aligned}$$

i.e. the sensitivity is a proportionality factor between a small relative perturbation in the plant's transfer function and the corresponding relative variation of the closed-loop transfer function. If you are interested in the subtleties of *this* kind of robustness, you can refer to the classical textbook Doyle, Francis, and Tannenbaum, *Feedback Control Theory.*

The closed loop is called *internally stable* if all the four transfer functions $W_{\text{cl}}(z)$, $S(z)$, $S_m(z)$, and $S_a(z)$ are BIBO-stable. Note the subtlety: *external stability* of the closed-loop would mean just the BIBO-stability of $W_{\text{cl}}(z)$, and this is not equivalent to *internal* stability, because a bad choice of either $q(z)$ or $p(z)$ could make $W_{\text{cl}}(z)$ stable but one of the other three functions unstable.

Example. Let $W(z) = \frac{1/4}{(z-1)(z-2)}$ and $C(z) = \frac{z-2}{z}$. Then

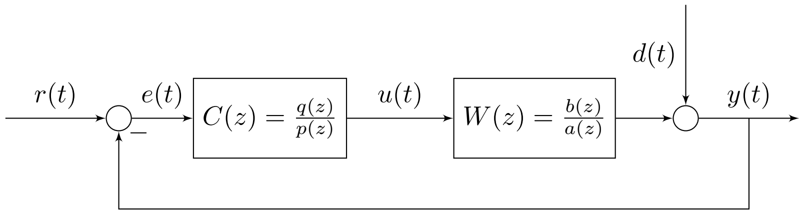
$$W_{\text{cl}}(z) = \frac{C(z)W(z)}{1+C(z)W(z)} = \frac{\frac{z-2}{z}\frac{1/4}{(z-1)(z-2)}}{1+\frac{z-2}{z}\frac{1/4}{(z-1)(z-2)}} = \frac{1/4}{z(z-1)+1/4} = \frac{1/4}{(z-1/2)^2}$$

is clearly BIBO-stable, and so are also $S(z)$ and $S_m(z)$; however,

$$S_a(z) = \frac{W(z)}{1+C(z)W(z)} = \frac{\frac{1/4}{(z-1)(z-2)}}{1+\frac{z-2}{z}\frac{1/4}{(z-1)(z-2)}} = \frac{\frac{z/4}{(z-2)}}{z(z-1)+1/4} = \frac{z/4}{(z-1/2)^2(z-2)}$$

is not BIBO-stable because of the pole at 2. Thus $C(z)$ attains external stability but not internal stability: canceling unstable poles in the naive way is not a good idea.

Luckily enough, internal stability actually calls for a unique requirement, because the denominator of all the four transfer functions is the same polynomial $a(z)p(z) + b(z)q(z)$, and this requirement is easy to satisfy by suitably designing the controller $q(z)/p(z)$. If we ensure that $a(z)p(z)+b(z)q(z)$ has roots in the open unit circle, and possibly *well-damped*, that is "well inside" the open unit circle $\{|z| < 1\}$, then the impact of disturbance is reduced, and in particular no disturbance can de-stabilize the closed loop. This said, to support intuition we consider a simplified scheme comprising only the reference signal and the load disturbance (keep in mind that the other disturbances are still there):



0.2 Stabilization

It goes without saying that internal stabilization calls for a Diophantine equation. Let $a_{\text{cl}}(z)$ be the desired common denominator of the four transfer functions to stabilize. Then

$$a(z)p(z) + b(z)q(z) = a_{\text{cl}}(z) \tag{1}$$

is a standard Diophantine equation. We assume that $\gcd(a, b) = 1$, i.e. that a and b are *coprime*: this corresponds to assuming that $W(z)$ is the transfer function of a reachable and observable system.

Since $\gcd(a, b) = 1$ we know that (1) has a solution p, q *for arbitrary* $a_{\text{cl}}(z)$; the solution is, however, not unique. The name of the game is now to put a bound on the degrees of p and q, more precisely to ensure that $C(z) = q(z)/p(z)$ is *proper*, so that it is the transfer function of a causal system that can be realized and implemented as a forward recursion.

The standard scheme goes as follows: let $\deg a = n$, $\deg b \leq n - 1$, and impose the degrees of p, q to be at most $n-1$. If now we impose also $\deg a_{\text{cl}} = 2n-1$, then the only possibility is that $\deg a_{\text{cl}} = \deg(ap) = \deg a+\deg p$, so that p has degree *exactly* $n - 1$, and $C(z)$ is proper. The polynomials p and q have n unknown coefficients each, and if $\deg a_{\text{cl}} = 2n - 1$, then equating each of its $2n$ coefficients with the coefficient of the corresponding power of z in the left-hand side of (1) we obtain a linear equation. Thus, (1) yields a linear system of $2n$ equations in $2n$ unknowns, that can wisely recast as Sylvester equation: and of course you remember that the Sylvester matrix is non-singular because $\gcd(a, b) = 1$, so that the solution is *unique*. Finally, the solution has also a nice look if we let, without loss of generality, a and a_{cl} be *monic*, because this forces p to be monic too.

The correspondence between degrees, number of unknowns and number of equations is resumed in the following table:

	$a(z)$	$\cdot$	$p(z)$	$+$	$b(z)$	$\cdot$	$q(z)$	$=$	$a_{\text{cl}}(z)$
degree	n		$n-1$		$\leq n-1$		$\leq n-1$		$2n-1$
unknowns			n				n		
equations									$2n$

There is a particular choice for closed-loop poles allowing us to attain a goal that is not possible in continuous-time control. If all the poles (counting multiplicity) of the closed loop are at the origin, that is if $a_{\text{cl}}(z) = z^{2n-1}$, then *the steady state corresponding to constant inputs is reached in finite time.*

This is better understood in terms of a state-space realization: the poles of the transfer function (of the closed-loop system) correspond to the eigenvalues of some dynamic matrix $A_{\text{cl}} \in \mathbb{R}^{N\times N}$. A suitable change of basis brings A_{cl} in an upper-triangular form $\bar{A}_{\text{cl}}$ with zeros on the diagonal (for example the canonical Jordan form); it follows that $\bar{A}_{\text{cl}}^N = 0$. Thus, in this basis, for all $t > N$ the state evolution corresponding to a constant reference $r(t) \equiv \bar{r}$ reads

$$\xi(t) = \bar{A}_{\text{cl}}^t \xi_0 + \sum_{\tau=0}^{t-1} \bar{A}_{\text{cl}}^{t-1-\tau} B r(\tau) = \left(\sum_{\tau=t-N}^{t-1} \bar{A}_{\text{cl}}^{t-1-\tau} B \right) \bar{r} = \left(\sum_{\tau=0}^{N-1} \bar{A}_{\text{cl}}^{\tau} B \right) \bar{r} \;=\; \text{constant.}$$

A discrete-time control scheme that allocates all the poles (or the eigenvalues) at the origin is called a *dead-beat controller*.

Example. Suppose that $W(z) = \frac{3z}{z^2-4}$. We design a controller $C(z) = \frac{q(z)}{p(z)}$ such that all the poles of the closed-loop transfer function[s] lie at the origin. Here $a(z) = z^2 - 4$, so $\deg a = n = 2$; we choose $\deg p = n - 1$, $\deg q \leq n - 1$, and $a_{\text{cl}}(z) = z^{2n-1} = z^3$. The Diophantine equation reads:

$$\begin{aligned} a_{\text{cl}}(z) &= a(z)p(z) + b(z)q(z); \\ z^3 &= (z^2 - 4)(p_1 z + p_0) + 3z(q_1 z + q_0); \\ z^3 &= (p_1)z^3 + (p_0 + 3q_1)z^2 + (-4p_1 + 3q_0)z + (-4p_0); \end{aligned}$$

equating coefficients we immediately find $p_1 = 1$ and $p_0 = 0$; then by substitution we recover $q_1 = 0$ and $q_0 = \frac{4}{3}$; the same solution is obtained solving the following Sylvester equation:

$$\left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 3 & 0 \\ -4 & 0 & 0 & 3 \\ 0 & -4 & 0 & 0 \end{array}\right] \left[\begin{array}{c} p_1 \\ p_0 \\ \hline q_1 \\ q_0 \end{array}\right] = \left[\begin{array}{cc|cc} \alpha_2 & & & \\ \alpha_1 & \alpha_2 & \beta_1 & \\ \alpha_0 & \alpha_1 & \beta_0 & \beta_1 \\ & \alpha_0 & & \beta_0 \end{array}\right] \left[\begin{array}{c} p_1 \\ p_0 \\ \hline q_1 \\ q_0 \end{array}\right] = \left[\begin{array}{c} c_3 \\ c_2 \\ c_1 \\ c_0 \end{array}\right] = \left[\begin{array}{c} 1 \\ 0 \\ 0 \\ 0 \end{array}\right].$$

Thus, the dead-beat controller reads $C(z) = \frac{4/3}{z}$.

0.3 Reference tracking and disturbance rejection

Reference tracking is the fact that

$$\lim_{t \to +\infty} r(t) - y(t) = \lim_{t \to +\infty} e(t) = 0. \tag{2}$$

The typical requirement is that (2) holds *at least* for constant references, i.e. for all references of the kind $r(t) = k\mathbb{1}(t)$.

The easiest way to put it down, according to me, is to recall that $e(t) = S(z)r(t)$. This is written in symbolical form (where $S(z)$ is meant as an operator between sequences), but a more orthodox way to write the same thing would be in terms of $\mathcal{Z}$-transforms: $E(z) = S(z)R(z)$. A constant reference has the transform $R(z) = \mathcal{Z}[k\mathbb{1}(t)] = \frac{kz}{z-1}$, hence, by the *final value theorem*,

$$\lim_{t \to +\infty} e(t) = \lim_{z \to 1} (z-1)E(z) = \lim_{z \to 1} (z-1)S(z)\frac{kz}{z-1} = k \cdot \lim_{z \to 1} S(z).$$

If we manage to obtain that $\lim_{z \to 1} S(z) = 0$, which is the case if $S(1) = 0$, then (2) is attained for all constant references. (Note that, since $S(z) + W_{\text{cl}}(z) = 1$, the requirement $S(1) = 0$ is equivalent to $W_{\text{cl}}(1) = 1$.) Now $S(z) = \frac{a(z)p(z)}{a(z)p(z)+b(z)q(z)}$: if properly designed, the denominator plays no role, and $a(z)$ is fixed. If, by a lucky coincidence, $a(z) = (z-1)a'(z)$ for some polynomial $a'(z)$, reference tracking is already enforced by $a(z)$ and then the stabilizing controller designed in the previous section is sufficient to satisfy the requirement; otherwise we must play with $p(z)$ and impose that it contains the factor $(z - 1)$.

Here is how: we let $p(z) = (z-1)\bar{p}(z)$ and $\bar{a}(z) = a(z)(z-1)$; the Diophantine equation becomes

$$\begin{aligned} &a(z)p(z) + b(z)q(z) = a(z)(z-1)\bar{p}(z) + b(z)q(z) = \\ &\bar{a}(z)\bar{p}(z) + b(z)q(z) = a_{\text{cl}}(z). \end{aligned} \tag{3}$$

This is more or less the same game as in (1), except that now $\bar{a}(z)$ contains the factor $(z - 1)$ by design, and $\deg \bar{a} = \deg a + 1$. Equation (4) can be solved increasing $\deg q = n - 1 + 1 = n$ and $\deg a_{\text{cl}} = 2n - 1 + 1 = 2n$;

one may think that $\deg \bar{p} = n$ is also required, but this is not the case, because the final controller will still be $C(z) = \frac{q(z)}{p(z)} = \frac{q(z)}{(z-1)\bar{p}(z)}$, i.e. it will remain proper also if we keep $\deg \bar{p} = n-1$. The situation is resumed in the following table:

	$a(z)$	$\cdot$	$(z-1)$	$\cdot$	$\bar{p}(z)$	$+$	$b(z)$	$\cdot$	$q(z)$	$=$	$a_{\text{cl}}(z)$
degree	n		1		$n-1$		$\leq n-1$		$\leq n$		$2n$
unknowns					n				$n+1$		
equations											$2n+1$

The final sensitivity function is

$$S(z) = W_{r\to e}(z) = \frac{a(z)p(z)}{a(z)p(z) + b(z)q(z)} = \frac{(z-1)a(z)\bar{p}(z)}{a(z)p(z) + b(z)q(z)}$$

and since $S(1) = 0$, it attains reference tracking for any constant reference.

The requirement that $(z-1)$ be present among the factors of $p(z)$ is often called *forcing integration in the loop*. Indeed by forcing a term $(z-1)$ in $p(z)$ we have *de facto* forced a term $\frac{1}{z-1}$ in the *open-loop* transfer function:

$$C(z)W(z) = \frac{q(z)}{p(z)}W(z) = \frac{1}{(z-1)}\frac{q(z)}{\bar{p}(z)}W(z).$$

Now $\frac{1}{z-1}$ is the transfer function of a "discrete-time integrator", that is an *adder*:

$$\begin{aligned}
y(t) &= \frac{1}{z-1}u(t) \\
(z-1)y(t) &= u(t) \\
y(t+1) &= y(t) + u(t), \qquad \text{and starting e.g. from } y(0) = 0 \\
y(t) &= \sum_{\tau=0}^{t-1} u(\tau).
\end{aligned}$$

At this point you should recall from your basic control courses that to attain asymptotic reference tracking of constant references *in continuous time* one has to impose the presence of an integrator $\int$, that is of a factor $\frac{1}{s}$, in the open-loop transfer function. This is precisely the same technique.

It shouldn't come as a shock that the closed loop with the controller obtained from (4) attains also *load disturbance rejection*, that is: *in the long run a constant disturbance $d(t) = h\mathbb{1}(t)$ has no effect on the output.* On one hand this is fairly intuitive, because any constant disturbance must be rejected, if constant reference tracking has to be attained. On the other hand one may notice that

$$S(z) = W_{r\to e}(z) = \frac{1}{1 + C(z)W(z)} \equiv W_{d\to y}(z),$$

so that the response $y_d(t) = W_{d\to y}(z)d(t)$ to any constant load disturbance satisfies

$$\lim_{t\to+\infty} y_d(t) = \lim_{z\to 1}(z-1)W_{d\to y}(z)D(z) = \lim_{z\to 1}(z-1)S(z)\frac{hz}{z-1} = hS(1) = 0.$$

Example. Suppose again that $W(z) = \frac{3z}{z^2-4}$. We design a controller $C(z) = \frac{q(z)}{p(z)}$ such that all the poles of the closed-loop transfer function[s] lie at the origin, and moreover we require integral action. The Diophantine equation to solve reads

$$a_{\text{cl}}(z) = \overbrace{a(z)(z-1)}^{\bar{a}(z)}\bar{p}(z) + b(z)q(z);$$

Here $\bar{a}(z) = (z^2-4)(z-1) = z^3 - z^2 - 4z + 4$ so that $\deg \bar{a} = \deg a + 1 = n + 1 = 3$, $\deg q = \deg \bar{a} - 1 = 2$, $\deg \bar{p} = \deg p - 1 = 1$, and $a_{\text{cl}}(z) = z^{2n} = z^4$. We proceed equating coefficients:

$$\begin{aligned}
z^4 &= (z^3 - z^2 - 4z + 4)(\bar{p}_1 z + \bar{p}_0) + 3z(q_2 z^2 + q_1 z + q_0); \\
z^4 &= (\bar{p}_1)z^4 + (-\bar{p}_1 + \bar{p}_0 + 3q_2)z^3 + (-4\bar{p}_1 - \bar{p}_0 + 3q_1)z^2 + (4\bar{p}_1 - 4\bar{p}_0 + 3q_0)z + (4\bar{p}_0);
\end{aligned}$$

equating coefficients we immediately find $\bar{p}_1 = 1$ and $\bar{p}_0 = 0$; the remaining equations lead to $q_2 = \frac{1}{3}$, $q_1 = \frac{4}{3}$ and $q_0 = -\frac{4}{3}$. The same solution is obtained by solving the following Sylvester equation:

$$\left[\begin{array}{cc|ccc} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 3 & 0 & 0 \\ -4 & -1 & 0 & 3 & 0 \\ 4 & -4 & 0 & 0 & 3 \\ 0 & 4 & 0 & 0 & 0 \end{array}\right] \left[\begin{array}{c} \bar{p}_1 \\ \bar{p}_0 \\ \hline q_2 \\ q_1 \\ q_0 \end{array}\right] = \left[\begin{array}{cc|ccc} \bar{\alpha}_3 & & & & \\ \bar{\alpha}_2 & \bar{\alpha}_3 & \beta_1 & & \\ \bar{\alpha}_1 & \bar{\alpha}_2 & \beta_0 & \beta_1 & \\ \bar{\alpha}_0 & \bar{\alpha}_1 & & \beta_0 & \beta_1 \\ & \bar{\alpha}_0 & & & \beta_0 \end{array}\right] \left[\begin{array}{c} \bar{p}_1 \\ \bar{p}_0 \\ \hline q_2 \\ q_1 \\ q_0 \end{array}\right] = \left[\begin{array}{c} c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{array}\right] = \left[\begin{array}{c} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{array}\right].$$

Thus $\bar{p}(z) = \bar{p}_1 z + \bar{p}_0 = z$, and the dead-beat controller attaining integral action reads

$$C(z) = \frac{q(z)}{\underbrace{\bar{p}(z)(z-1)}_{p(z)}} = \frac{\frac{1}{3}z^2 + \frac{4}{3}z - \frac{4}{3}}{z(z-1)}.$$

0.4 Reference tracking for more complex reference signals

To write...

Goal: attain $\lim_{t\to+\infty} e(t) = 0$ for more complex references $r(t)$.

$E(z) = S(z)R(z)$.

Look at the denominator in the transform $R(z)$

signal	**transform**	**convergence region**
$\mathbb{1}(t)$	$\frac{z}{z-1}$	$\lvert z\rvert > 1$
$\mathbb{1}(t)\cdot t$	$\frac{z}{(z-1)^2}$	$\lvert z\rvert > 1$
$\mathbb{1}(t)\cdot t^2$	$\frac{z(z-1)}{(z-1)^3}$	$\lvert z\rvert > 1$
$\mathbb{1}(t)\cdot\cos(\omega t)$	$\frac{z(z-\cos\omega)}{z^2-2z\cos\omega+1}$	$\lvert z\rvert > 1$
$\mathbb{1}(t)\cdot\sin(\omega t)$	$\frac{z\sin\omega}{z^2-2z\cos\omega+1}$	$\lvert z\rvert > 1$

Denominator $= k(z)$, $\deg k = n_k$. Force the factor $k(z)$ in $p(z)$ so that they cancel out and:

$$\begin{aligned}\lim_{t\to+\infty} e(t) &= \lim_{z\to1}(z-1)S(z)\frac{n(z)}{k(z)} \\ &= \lim_{z\to1}(z-1)\frac{a(z)\overbrace{k(z)\bar{p}(z)}^{p(z)}}{a(z)p(z)+b(z)q(z)}\frac{n(z)}{k(z)} \\ &= \lim_{z\to1}(z-1)\frac{a(z)\bar{p}(z)}{a(z)p(z)+b(z)q(z)}n(z) \\ &= 0.\end{aligned}$$

Let $p(z) = k(z)\bar{p}(z)$ and $\bar{a}(z) = a(z)k(z)$; the Diophantine equation becomes

$$\begin{aligned}&a(z)p(z) + b(z)q(z) = a(z)k(z)\bar{p}(z) + b(z)q(z) = \\ &\bar{a}(z)\bar{p}(z) + b(z)q(z) = a_{\text{cl}}(z).\end{aligned} \tag{4}$$

Degrees 'n stuff:

	$a(z)$	$\cdot$	$k(z)$	$\cdot$	$\bar{p}(z)$	$+$	$b(z)$	$\cdot$	$q(z)$	$=$	$a_{\text{cl}}(z)$
degree	n		n_k		$n-1$		$\leq n-1$		$\leq n-1+n_k$		$2n-1+n_k$
unknowns					n				$n+n_k$		
equations											$2n+n_k$

Tracking ramps and so on: the power of $(z-1)^m$ in $k(z)$ (or anyway in the open-loop t.f.) is called the *type* of the loop. Same story as in Fondamenti di Automatica.

Exercise: $W(z) = \frac{z-1/3}{(z-1/2)(z+2)}$. Unstable.
Write code: Solve Diophantine equation to stabilize.
Solve Diophantine equation to track constant references.
Solve Diophantine equation to track sinusoids $r(t) = \sin(\omega t + \phi)$, where $\omega \in [-\pi, \pi]$, e.g. $\omega = 1/10$.
Note: $\sin(\omega t + \phi) = \sin(\omega t)\cos(\phi) + \cos(\omega t)\sin(\phi) = a\sin(\omega t) + b\cos(\omega t)$. Don't mind about the phase, the transforms of sin and cos have the same denominator, so the regulator will track any linear combination.
Simulate.

0.5 Forcing zero cancellations

Recall that the zeros of the plant, i.e. the roots of $b(z)$, are zeros also of the closed-loop transfer function $W_{\text{cl}}(z) = \frac{b(z)q(z)}{a(z)p(z)+b(z)q(z)}$. Sometimes it is desirable to design the controller in such a way that some, or all, of such zeros are canceled. For example, let $b(z) = b_c(z)b_u(z)$, where the roots of $b_c(z)$ are those that we want to get rid of. The goal is attained if we impose

$$a_{\text{cl}}(z) = b_c(z)a'_{\text{cl}}(z),$$

because the closed loop becomes $W_{\text{cl}}(z) = \frac{b_c(z)b_u(z)q(z)}{b_c(z)a'_{\text{cl}}(z)} = \frac{b_u(z)q(z)}{a'_{\text{cl}}(z)}$. It is immediate to check that, in order to ensure that the Diophantine equation

$$a(z)p(z) + b_c(z)b_u(z)q(z) = b_c(z)a'_{\text{cl}}(z) \tag{5}$$

is solvable, this requirement imposes in turn that $b_c(z)$ divides $p(z)$, i.e. that $p(z) = b_c(z)p'(z)$ for some polynomial $p(z) = b_c(z)p'(z)$. With this requirement, (5) becomes

$$a(z)p'(z) + b_u(z)q(z) = a'_{\text{cl}}(z), \tag{6}$$

which is a standard Diophantine equation, whose solution we can restrict and solve with the techniques of the previous sections. The final controller, attaining the zero cancellation, is $C(z) = \frac{q(z)}{p(z)} = \frac{q(z)}{b_c(z)p'(z)}$.

But there is a big *caveat* here: any effect of the zeros in $b_c(z)$ gets canceled from $W_{\text{cl}}(z)$, $S(z)$, and $S_a(z)$, that become respectively

$$W_{\text{cl}}(z) = \frac{b_u(z)q(z)}{a'_{\text{cl}}(z)}, \quad S(z) = \frac{a(z)p'(z)}{a'_{\text{cl}}(z)}, \quad S_a(z) = \frac{b_u(z)p(z)}{a'_{\text{cl}}(z)}.$$

But all the zeros in $b_c(z)$ become *poles* of

$$S_m(z) = \frac{a(z)q(z)}{a(z)p(z)+b(z)q(z)} = \frac{a(z)q(z)}{a_{\text{cl}}(z)} = \frac{a(z)q(z)}{b_c(z)a'_{\text{cl}}(z)}.$$

As a consequence, *you cannot cancel unstable zeros, or the closed-loop system will become internally unstable.* Indeed, the common practice is to cancel only stable and *well-damped* zeros (i.e. roots of $b(z)$ close enough to the origin).

0.6 A controller with two degrees of freedom

The controller with which we have been dealing so far has the structure $u(t) = C(z)e(t) = \frac{q(z)}{p(z)}e(t)$, that is

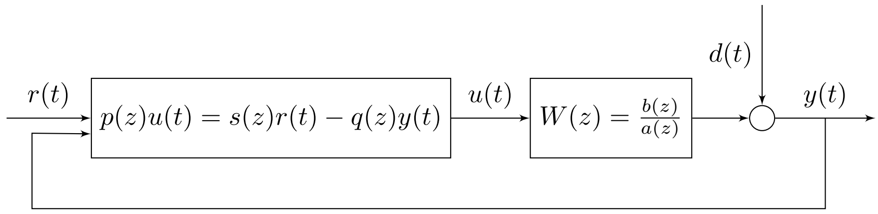
$$\begin{aligned} p(z)u(t) &= q(z)e(t) = q(z)(r(t) - y(t)) \\ &= q(z)r(t) - q(z)y(t). \end{aligned} \tag{7}$$

As we have seen, it attains both asymptotic reference tracking and disturbance rejection, *of the same class of signals.* This is called a controller with *one degree of freedom.*

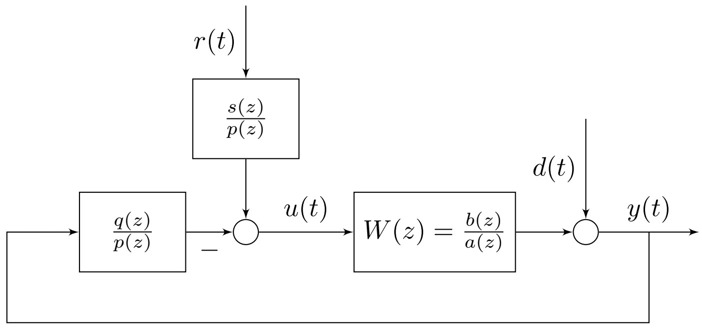
A "more realistic" design, as Åström and Wittenmark put it, is to decouple the two problems: for example we could want to attain reference tracking for some class of signals (e.g. constants) and reject load disturbances of another class (e.g. sinusoids of a given frequency), while keeping under control the degrees of the involved polynomials and maintaining the design simple enough. This can be attained with a *two-degrees-of-freedom controller*:

$$p(z)u(t) = s(z)r(t) - q(z)y(t). \tag{8}$$

Equation (8), which is of course a generalization of (7), is represented in the following diagram:



Since in operator notation (8) can be rewritten as $u(t) = \frac{s(z)}{p(z)}r(t) - \frac{q(z)}{p(z)}y(t)$, the following is an equivalent diagram:



We obtain, as we had in the previous sections,

$$S(z) = W_{d\to y}(z) = \frac{1}{1 + \frac{b(z)}{a(z)}\frac{q(z)}{p(z)}} = \frac{a(z)p(z)}{a(z)p(z) + b(z)q(z)},$$

therefore we will use a suitable Diophantine equation to stabilize the closed loop and attain disturbance rejection (the roots of $p(z)$ must cancel the poles of the transform of the disturbances to reject): this is the "first degree of freedom". On the other hand, we have

$$W_{\text{cl}}(z) = W_{r\to y}(z) = \frac{\frac{b(z)}{a(z)}\frac{s(z)}{p(z)}}{1 + \frac{b(z)}{a(z)}\frac{q(z)}{p(z)}} = \frac{b(z)s(z)}{a(z)p(z) + b(z)q(z)};$$

while the denominator is always the same, the numerator is different from before: we can play with $s(z)$ in order to impose, to some extent, the zero structure of the closed-loop transfer function; this is the "second degree of freedom". It should be clear that the bare minimum is to attain $W_{\text{cl}}(1) = 1$ (constant reference tracking): this becomes immediately available by choosing $s(z) =$ the constant $\frac{a(1)p(1)+b(1)q(1)}{b(1)}$; but more can be done.

An ideal goal would be to impose the closed-loop transfer function altogether, i.e. to attain

$$W_{\text{cl}}(z) = \frac{b(z)s(z)}{a_{\text{cl}}(z)} := \frac{b_m(z)}{a_m(z)}$$

for some prescribed polynomials $b_m(z)$, $a_m(z)$. In other words, to impose both the pole-structure and the zero-structure of the closed-loop system. As was the case in Section 0.5, this is in general impossible, because it would mean to cancel *all* the zeros of the plant (the roots of $b(z)$), and exactly as before this would cause the closed loop to be internally unstable. Therefore the zero-structure can be changed only to some extent; in particular *all the unstable zeros must remain.*

Here follows the design procedure (the remainder of this section is just to provide some insight and is not exhaustive at all; if you are interested in all the tricky details you can refer to Åström and Wittenmark, *Adaptive Control*).

First, we factor $b(z) = b_c(z)b_u(z)$, where the zeros $b_c(z)$ are going to be canceled and the (unstable or poorly damped) zeros in $b_u(z)$ are going to remain. As we did in Section 0.5, we factor $a_{\mathrm{cl}}(z) = b_c(z)a'_{\mathrm{cl}}(z)$ accordingly:

$$W_{\mathrm{cl}}(z) = \frac{b(z)s(z)}{a_{\mathrm{cl}}(z)} = \frac{b_c(z)b_u(z)s(z)}{b_c(z)a'_{\mathrm{cl}}(z)} = \frac{b_u(z)s(z)}{a'_{\mathrm{cl}}(z)}.$$

That the zeros in $b_u(z)$ *are going to remain* means that $b_u(z)$ must divide the desired numerator $b_m(z)$, i.e. that $b_m(z) = b_u(z)b'_m(z)$ for some polynomial $b'_m(z)$. Now $a'_{\mathrm{cl}}(z)$ must be the product of the desired denominator $a_m(z)$ and some other polynomial, say $a_o(z)$, of high enough degree such that the forthcoming Diophantine equation is solvable:

$$W_{\mathrm{cl}}(z) = \frac{b_u(z)s(z)}{a_o(z)a_m(z)};$$

if now we choose $s(z) := a_o(z)b'_m(z)$ we obtain, as desired,

$$W_{\mathrm{cl}}(z) = \frac{b_u(z)\overbrace{a_o(z)b'_m(z)}^{s(z)}}{\underbrace{a_o(z)a_m(z)}_{a'_{\mathrm{cl}}(z)}} = \frac{b_u(z)b'_m(z)}{a_m(z)} = \frac{b_m(z)}{a_m(z)}.$$

It remains to setup the Diophantine equation. Check the correspondences:

$$\begin{aligned} a(z)p(z) + b(z)q(z) &= a_{\mathrm{cl}}(z), \\ a(z)p(z) + b_c(z)b_u(z)q(z) &= b_c(z)a'_{\mathrm{cl}}(z), \qquad \text{(zero cancellation: } p(z) = b_c(z)p'(z) \text{ for some } p'(z)) \\ a(z)p'(z) + b_u(z)q(z) &= a'_{\mathrm{cl}}(z) = a_o(z)a_m(z), \end{aligned}$$

which is equation (6) where $b_u(z)$ contains just the unstable/poorly damped zeros and $a'_{\mathrm{cl}}(z)$ has been properly factorized in order to pursuit model matching. Disturbance rejection can now be attained imposing that $p'(z) = \bar{p}(z)k(z)$ for a suitable $k(z)$ canceling the disturbances, and proceeding as in Sections 0.3, 0.4.

0.7 The final control unit

Here is how the prototypical control unit of a self-tuning regulator works.

In the first stage of the design of the self-tuning regulator, *a parametric model class has been chosen by the designer* as a family of models that likely fits the plant and wishfully contains the "true" parameter $\bar{\theta}$. To fix ideas, let the model class be ARX(2,2): "*AutoRegressive* process with eXhogenous input, 2 delays in the autoregressive part and 2 delays in the input part". This means the class of models with the following structure:

$$y(t) = \underbrace{a_1y(t-1) + a_2y(t-2)}_{\text{autoregressive part}} + \underbrace{b_1u(t-1) + b_2u(t-2)}_{\text{input part}} \quad [+ \text{ noise}(t)]. \tag{9}$$

The noise term is typical of the models used in system identification; it means an i.i.d. sequence of *random variables*, adding information to the process $\{y(t)\}$. Feel free to ignore it for now.

The job of the *tuning* unit is to provide, robustly with respect to the possible presence of noise, an estimate $\hat{\theta} = (\hat{a}_1, \hat{a}_2, \hat{b}_1, \hat{b}_2)$ of the "true" parameter: such estimate typically improves in time, so it *depends on* time, and is actually a $\hat{\theta}_t$. Let $\hat{\theta} = \hat{\theta}_t$ be the current estimate: the control unit receives $\hat{\theta}$ from the tuning unit.

The control unit is designed to compute a control action depending on some past samples $y(t-1), y(t-2), u(t-1), u(t-2)$, etc.; so it stores them in some way and behaves like a recursive algorithm. The model class for which it can compute a control action is

$$y(t) = a_1y(t-1) + a_2y(t-2) + b_1u(t-1) + b_2u(t-2),$$

where $\theta = (a_1, a_2, b_1, b_2)$ is the unknown parameter: for every $\theta \in \mathbb{R}^4$, the control unit is supposed to provide a recursive control algorithm.

At each time step, or every T time steps (this is a designer's choice), it substitutes the unknown parameters with the current estimate received by the tuning unit:

$$y(t) = \hat{a}_1 y(t-1) + \hat{a}_2 y(t-2) + \hat{b}_1 u(t-1) + \hat{b}_2 u(t-2). \tag{10}$$

(Recall: the idea of employing the estimate $\hat{\theta} = \hat{\theta}_t$ *as if it was* the "true" parameter $\bar{\theta}$ is called the *certainty equivalence principle.*) Now, with a bit of manipulations, from (10) we obtain

$$\begin{aligned}
y(t) &= \hat{a}_1 z^{-1} y(t) + \hat{a}_2 z^{-2} y(t) + \hat{b}_1 z^{-1} u(t) + \hat{b}_2 z^{-2} u(t);\\
(1 - \hat{a}_1 z^{-1} - \hat{a}_2 z^{-2}) y(t) &= (\hat{b}_1 z^{-1} + \hat{b}_2 z^{-2}) u(t);\\
y(t) &= \frac{\hat{b}_1 z^{-1} + \hat{b}_2 z^{-2}}{1 - \hat{a}_1 z^{-1} - \hat{a}_2 z^{-2}} u(t) = \frac{\hat{b}_1 z + \hat{b}_2}{z^2 - \hat{a}_1 z - \hat{a}_2} u(t)\\
&:= \frac{\beta_1 z + \beta_0}{z^2 + \alpha_1 z + \alpha_0} u(t) = \frac{b(z)}{a(z)} u(t) = \hat{W}(z) u(t) \qquad \text{(note that } a(z) \text{ is monic).}
\end{aligned}$$

$\hat{W}(z)$ is the current model of the plant used by the control unit. It comes as a strictly proper rational transfer function having $\deg a = n = 2$. We assume that a and b are, at every update of the parameter estimate, *always* coprime: otherwise the update should be rejected, but if it happens by construction, or too often, it probably means that the model class was over-parameterized ($\deg a$ and $\deg b$ are too high), and the whole design should be reconsidered.

Each time the model is updated according to $\hat{\theta} = \hat{\theta}_t$, the control unit resets its control scheme (say with the method of Section 0.3, enforcing integral action, solving a Diophantine equation etc.):

$$\begin{aligned}
C(z) &= \frac{q(z)}{p(z)} = \frac{q_2 z^2 + q_1 z + q_0}{(z-1)(z - \bar{p}_0)} = \frac{q_2 z^2 + q_1 z + q_0}{z^2 + p_1 z + p_0} \qquad \text{(choose } a_{\text{cl}}(z) \text{ s.t. } p(z) \text{ is also monic)}\\
&= \frac{q_2 + q_1 z^{-1} + q_0 z^{-2}}{1 + p_1 z^{-1} + p_0 z^{-2}};\\
(1 + p_1 z^{-1} + p_0 z^{-2}) u(t) &= (q_2 + q_1 z^{-1} + q_0 z^{-2}) e(t)\\
u(t) &= -p_1 z^{-1} u(t) - p_0 z^{-2} u(t) + q_2 e(t) + q_1 z^{-1} e(t) + q_0 z^{-2} e(t)\\
&= -p_1 u(t-1) - p_0 u(t-2) + q_2 e(t) + q_1 e(t-1) + q_0 e(t-2).
\end{aligned}$$

To compute the control action, the control unit must have stored two past samples $e(t-1) = r(t-1) - y(t-1)$ and $e(t-2) = r(t-2) - y(t-2)$. Contextually with the update of $\hat{\theta} = \hat{\theta}_t$, also the samples $r(t)$ and $y(t)$ come, hence $e(t) = r(t) - y(t)$ becomes available and the control unit can finally output $u(t)$.

The timing and the update sequence of the whole closed loop are resumed in the following table:

at time	**the plant**	**the tuning unit**	**the control unit**
t	has memory (as a state) of: $u(t-1), u(t-2), \ldots$ $y(t-1), y(t-2), \ldots$ has evolved: the current output is $y(t)$	has memory (state) of: $u(t-1), u(t-2), \ldots$ $y(t-1), y(t-2), \ldots$ $\hat{\theta}_{t-1}, [\ldots]$ measures $y(t)$ from plant; updates estimate: $\hat{\theta}_t$	has memory (state) of: $u(t-1), u(t-2), \ldots$ $e(t-1), e(t-2), \ldots$ $\hat{\theta}_{t-1}$, corresponding control law $C(z)$ receives $\hat{\theta}_t$ from the tuning unit, updates the control law $C(z)$, receives $r(t)$ from the operator, measures $y(t)$ from the plant, computes $e(t)$ and actuates $u(t)$
t	receives $u(t)$ from control unit; can evolve: next output will be $y(t+1)$	reads $u(t)$ from control unit	
$t+1$	has memory of: $u(t), u(t-1), \ldots$ $y(t), y(t-1), \ldots$ has evolved: the current output is $y(t+1)$	has memory of: $u(t), u(t-1), \ldots$ $y(t), y(t-1), \ldots, \hat{\theta}_t, [\ldots]$ measures $y(t+1)$; updates estimate: $\hat{\theta}_{t+1}$	has memory of: $u(t), u(t-1), \ldots, e(t), e(t-1), \ldots$ $\hat{\theta}_t$, corresponding control law $C(z)$ receives $\hat{\theta}_{t+1}$ and updates control law, receives $r(t+1)$, $y(t+1)$; actuates $u(t+1)$
$t+1$	receives $u(t+1)$, can evolve: next output will be $y(t+2)$	reads $u(t+1)$	
$t+2$	and so on...	and so on...	and so on...

//

This concludes our discussion about linear control design.
We can now pass to the subject of parameter estimation, that is to the *tuning unit*.