Energy-Oriented Design, Control and Optimization of Robotic Systems

Synopsis of Results from NSF Grant #1536035

Hanz Richter, PhD
Professor, Mechanical Engineering Department
Cleveland State University

Modeling - Stored Energy Dynamics

Assume that the star configuration is used (all semiactive drives connected in parallel to the storage element). In [1] the following differential equation is derived for the currents to/from the storage element contributed by semiactive drive j:

$$i_j = \frac{r_j}{R_j}\left(a_j \dot{q}_j - r_j V_s\right) \tag{1}$$

Recall that $r_j = v_{jth-motor}/V_s$ is the control input, and that the augmented robot model has inputs $u_i = \frac{a_j r_j}{R_j} V_s$ for semiactive joints:

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + \mathcal{R}(q,\dot{q}) + g + \mathcal{T} = u \tag{2}$$

System dynamics for the complete system (robot and drive) include Eq. 2 and the dynamics of the charge y in the storage element. Due to the parallel connection we have $\dot{y} = \sum i_j$, which gives the required charge dynamics. But V_s is still unspecified.

Independence from storage element model

When working with supercapacitors, we could use simple capacitance models $\dot{y} = C\dot{V}_s$, and this is done in some of our publications. We could also use more realistic models involving several capacitance elements and internal resistances, or even fractional-order models for simulations.

But it turns out that we can completely avoid using any such model, at least for two purposes:

- To design controllers for the augmented robot: we use V_s as feedback
- To monitor and optimize energy utilization: we develop energy balance equations that hold exactly and don't need a model of the storage element.

The first aspect is dealt with an approach we call *semiactive virtual control*, or SVC. It simply involves regarding $u_i = \frac{a_j r_j}{R_j} V_s$ as a control input that can be directly specified (a virtual control). The final control r_j is then obtained by solving for r_j, and this involves dividing by V_s, which is assumed to be available from a sensor. This simple idea was first proposed in [3] and it goes a long way in facilitating energy-oriented robot control at analysis and practical levels. It is discussed in the next synopsis document.

The use of a specific virtual control law impacts how energy is distributed in the system in the form of work, storage and losses. This is captured by the second aspect above.

Energy Balance Equations

Suppose that a virtual feedback law $u = \tau^d(q, \dot{q})$ has been designed for the augmented robot. Solving for r_j gives

$$r_j = \frac{R_j \tau_j^d}{a_j V_s}$$

We can substitute this into Eq. 1 to find the currents as a function of virtual instead of actual controls. Then, the equation can be multiplied by V_s left and right and integrated to give the *internal energy balance equation*:

$$\Delta E_s = \int_{t_1}^{t_2} \sum_{j=1}^{e} \left(\tau_j^d \dot{q}_j - \frac{R_j}{a_j^2} \left(\tau_j^d \right)^2 \right) dt \tag{3}$$

where e is the number of semiactive drives and ΔE_s gives the change in the energy storage element between any two arbitrary times.

The above is a storage-centric balance. An *external energy balance* equation captures the energy behavior of the entire system. The derivation, independent of storage model is best presented in [1].

$$W_{act} = W_{ext} + \Delta E_m^T + \Sigma_m^T + \Delta E_s + \Sigma_e \tag{4}$$

The above includes the work of the active joints, W_{act} and the work introduced by external forces or moments, W_{ext}. The other terms are the changes in potential and kinetic energies of the augmented robot, ΔE_m, the change of stored electric energy ΔE_s and mechanical and electric losses Σ_m and Σ_e, respectively.

The integrand in the IEB is quadratic in virtual control and velocity (but not positive definite!). It is used as the basis for optimization. The EEB is just energy conservation, and it serves two purposes: i. it implicitly defines an energy conversion efficiency, and ii., it is an error-checking method, because it gives a second way to calculate ΔE_s.

Direct model inversion

In motion planning problems we might want to find inputs $r_j(t)$ that produce a specific trajectory $q(t)$ and also evaluate the resulting EEB and IEB. This is possible through a method we called "u-inversion" (because r was called u in that paper) presented for a 1 d.o.f. electromechanical system in [2] where the primary control problem was impedance regulation. The same idea applies to larger robotic systems as discussed here. A closely-related method was used in [4] to co-optimize the control and design parameters of a powered prosthetic leg using energy regeneration.

References

[1] P. Khalaf. *Design, Control, and Optimization of Robots with Advanced Energy Regenerative Drive Systems.* PhD thesis, Cleveland State University, 2019.

[2] Hanz Richter and Dhipak Selvaraj. Impedance control with energy regeneration in advanced exercise machines. In *American Control Conference (ACC), 2015*, pages 5890–5895. IEEE, 2015.

[3] Hanz Richter, Dan Simon, and Antonie van den Bogert. Semiactive virtual control method for robots with regenerative energy-storing joints. In *Proc. 19th IFAC World Congress, Cape Town, South Africa*, 2014.

[4] Holly E Warner. Optimal design and control of a lower-limb prosthesis with energy regeneration. Master's thesis, Cleveland State University, 2015.