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

Semiactive Virtual Control

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:

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + \mathcal{R}(q,\dot{q}) + g + \mathcal{T} = u \tag{1}$$

We would like to control the robot to perform a primary motion task, like tracking a desired trajectory, regulating the impedance relative to external forces/moments $\mathcal{T}$, controlling interaction force, or something else. At the same time, we want the primary task to be completed in an energy-efficient way.

For most of our work we have interpreted "energy-efficient" as maximizing the energy change in the storage element ΔE_s, or equivalently minimizing the energy consumption $-\Delta E_s$.

Semiactive virtual control, or SVC, 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.

If we compare this with backstepping control, the first step is the same: finding a good virtual control law. In backstepping, the next process is to force the virtual control and actual controls (which are dynamically related) to converge to each other, maintaining stability. This is done by extending the Lyapunov function associated with the virtual design with a term which is positive-definite on the error between virtual and actual controls.

In SVC, the second step is "zero order", because virtual and actual controls are algebraically related, trivializing the process, at least for nominal models. The virtual matching equations are just:

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

Upon this, τ^d becomes effective for the augmented robot, and the criteria under which τ^d was designed (robustness, stability, etc) applies to the actual system. In practice, three aspects must be observed:

1. Uncertain parameters: R_j and $n_j = \alpha_j n_j$ may be approximately known. Then there is a mismatch between τ^d as designed and the actual system input. Some analysis of this is presented in [2], indicating that parametric uncertainties of this kind show up as a disturbance term that can be compensated by a robust virtual design.

2. Saturation: r_j is a converter ratio (duty ratio), which is constrained to $[-1, 1]$ in many practical regenerative (4 quadrant) units. If the right-hand side term above evaluates to something outside this interval, the converter will saturate and there will be a mismatch between virtual and actual controls. An easy way to fix this is to use a higher overall V_s in the storage element.

3. $V_s \neq 0$: In numerical optimization, the algorithm may explore trajectories that completely discharge the storage element, leading to $V_s = 0$. This must be avoided by using constraints or using a sufficiently high initial energy in the storage element.

If inductance is included in the drive, the matching process is no longer simple. A partial answer to this case is through real-time measurements of the voltage across the inductor, as shown in [2]. If the inductance is inherent to the motor, no measurement is possible and estimation methods could be used. If there's an intentionally-placed inductor separate from the motor, then its voltage drop could be directly sensed and used for feedback. This aspect has not been sufficiently explored. In [4] we discuss a current-based matching law as an alternative to the above equation, but it presents a singularity. This is another aspect that must be re-examined.

Amin Ghorbanpour has extended SVC for applicability to motors with AC commutation, like brushless DC motors [1]. The matching equations are again different.

In hydraulic systems, the matching equations are different than the above, don't admit a direct solution and over-actuation may be present. The matching process must be done through optimization, see [3].

Example simulation

In the attached code, we perform trajectory control of a planar 3-link robot using a virtual controller based on inverse dynamics. No external forces are used, there are no uncertainties, and saturation is not included. The code shows how to simulate the system and perform both internal and external energy balances. It corresponds to the code used for paper [4].

Without saturation and a and R exactly known, the virtual matching equation and the right-hand-side term $\frac{a_j r_j}{R_j} V_s$ used in the plant block cancel exactly in simulation, which would appear silly. However, in a real-time laboratory implementation, the matching equation is performed in the control code, while the right-hand-side term is just a model of what happens in the physical system, and such exact cancellation does not occur.

Experience with real-time SVC implementations has been very good, as described in other synopsis documents.

References

[1] Amin Ghorbanpour and Hanz Richter. Control with optimal energy regeneration in robot manipulators driven by brushless dc motors. In *ASME 2018 Dynamic Systems and Control Conference*, pages V001T04A003–V001T04A003. American Society of Mechanical Engineers, 2018.

[2] Hanz Richter. A framework for control of robots with energy regeneration. *Journal of Dynamic Systems, Measurement, and Control*, 137(9):091004, 2015.

[3] Hanz Richter, Xin Hui, Antonie J van den Bogert, and Dan Simon. Semiactive virtual control of a hydraulic prosthetic knee. In *2016 IEEE Conference on Control Applications (CCA)*, pages 422–429. IEEE, 2016.

[4] 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.