

Computed Muscle Control

Theory

1.1 Overview: Computed Muscle Control (Residual Reduction)

What are CMC and RRA?

Both computed muscle control (CMC) and the residual reduction algorithm (RRA) are feedback-control based methods of estimating actuator controls such that actuator forces generate motion that tracks a desired trajectory (i.e. joint angles calculated from experimental data). Their similarity results in the algorithms sharing the same implementation (cmc.exe in OpenSim). They differ from standard feedback control methods in that the controls (forces) are solved via static optimization, which enables redundant actuation and allows for adjustments to the “desired” kinematics. CMC also accounts for the physiological delay in the generation of muscle forces.

In RRA, static optimization is utilized to slightly adjust kinematics (joint angles) to reduce the magnitude of residuals (i.e., non-physical forces applied to the model accounting for differences in measured forces and accelerations) from inverse dynamics. In a separate step the total model mass of the model and location of the COM of a targeted segment (typically the most massive, e.g. torso) are adjusted to eliminate offsets in residual forces and moments.

1.2 Theoretical background

1. We cast the inverse dynamics problem (joint torques from kinematics and external forces) as a tracking problem:
 - a. Unlike inverse dynamics, tracking requires a dynamical system
 - $\mathbf{M}\ddot{\mathbf{q}} = D(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{f}$, joint angles and velocities determined by integrating accelerations
 - b. Solve inverse system: \mathbf{f} is computed based on accelerations and states, but
 - accelerations are difficult to measure but positions are easy
 - double differentiation provides acceleration but also amplifies noise
 - instantaneous forces and continuity over time not ensured
 - what if we use feedback to get better estimates of accelerations?
 -

2. Feedback control:

1. (1) $\tilde{q} = \hat{q} - k_v(\dot{q} - \hat{\dot{q}}) - k_p(q - \hat{q})$ estimate acceleration using feedback
- (2) $f = \mathbf{M}\tilde{q} - D(q, \dot{q})$ compute controls (gen. forces, f)
- (3) integrate $\mathbf{M}\ddot{q} = D(q, \dot{q}) + f$

When actuator forces are torques, $f = \tau$, this is the method of computed torques.

2. For CMC, the production of torque is not instantaneous. The required torque at a given instant must be generated (in the case of muscles) over some prior time. To account for this, CMC changes the feedback law slightly, such that:

$$\tilde{q}(t+T) = \hat{q}(t+T) - k_v(\dot{q} - \hat{\dot{q}}) - k_p(q - \hat{q})$$

where the desired acceleration at small time in the future (T) is based on the experimental acceleration at the future time and the current position and velocity errors.

3. In the general case, “controls” (generalized forces, f) are computed from static optimization:

$$\text{Min: } J(x) = \sum_{i=1}^{nx} \left(\frac{f_i(x_i)}{f_i^{opt}} \right)^2 + \sum_{j=1}^{nq} w_{q_j} (\tilde{q}_j(t+T) - \ddot{q}_j(t+T))^2$$

$$\text{Subject to: } \mathbf{M}\ddot{q} = D(q, \dot{q}) + f(x)$$

RRA: Compute torques and residuals (τ, R) that track the desired kinematics.

$$\tau, R \in f(x) = F^{opt} x$$

Min:

$$J(\tau, R) = \sum_{i=1}^{n\tau} \left(\frac{\tau_i^{opt} x_i}{\tau_i} \right)^2 + \sum_{k=1}^6 w_{R_k} \left(\frac{R_k^{opt} x_k}{R_k} \right)^2 + \sum_{j=1}^{nq} w_{q_j} (\tilde{q}_j(t+T) - \ddot{q}_j(t+T))^2$$

$$\text{subject to: } \mathbf{M}\ddot{q} = F(q, \dot{q}) + \tau + R$$

Enables us to vary tracking behavior between minimizing the residuals and tracking the kinematics and is solved as a bounded (limits on torques) nonlinear optimization. Since torques have no dynamics, T can be small (e.g. $T = 0.001s$).

CMC: Controls (x) are now excitations and torque production now has muscle dynamics:

$$\tau = [\mathbf{A}(q)]f_m(a(x), l, q, \dot{q}) + r$$

Joint torques, τ , are now replaced by the product of a muscle moment arm matrix, \mathbf{A} , and muscle forces, f_m , which is a function of muscle activation, a , which in turn is a function of excitation, x , the controls. (NOTE: “reserve” torques (actuators), r , remain in case muscle forces alone are unable to achieve the necessary accelerations.

$$\text{Min: } J(\tilde{f}^m, x_{r,R}) = \sum_{i=1}^{nm} \left(\frac{\tilde{f}_i^m}{f_i^{opt}(\hat{q}, \hat{\dot{q}})} \right)^2 + \sum_{j=1}^{n\tau} \left(\frac{r_j^{opt} x_j}{r_j^{opt}} \right)^2 + \sum_{k=1}^6 \left(\frac{R_k^{opt} x_k}{R_k^{opt}} \right)^2 \Big|_{t+T}$$

Subject to: $\ddot{q}(t+T) = \tilde{q}(t+T)$ (using constraint is “fast target” in optimization)

$$\mathbf{M}\ddot{q} = F(q, \dot{q}) + \mathbf{A}f^m + r + R \Big|_{t+T}$$

$$f^{\min} \leq \tilde{f}^m \leq f^{\max} \Big|_{t+T}$$

Algorithm:

1. Set states: $\{a, l, q, \dot{q}\}$ at current time, t
2. Estimate future acceleration: $\tilde{q}(t+T) = \hat{q}(t+T) - k_v(\dot{q} - \hat{\dot{q}}) - k_p(q - \hat{q})$
 $T = 0.01s$ provides sufficient time for a muscle to reach a target force.
3. Integrate muscle dynamics only with $(\hat{q}, \hat{\dot{q}})$ from t to $t+T$, with $x_i = [0,1]$ to get bounds on muscle forces $(f_i^{\min}, f_i^{\max}) \Big|_{t+T}$
4. Solve static optimization to get the desired muscle forces, \tilde{f}_i^m , at $t+T$.
5. Root solve for x from desired muscle forces, given:
 $f^m(a(x), l, q, \dot{q}) = \tilde{f}_i^m \Big|_{t+T}$
6. Integrate muscle and skeletal (multibody) dynamics for states at $t+T$ and loop.

Adjusting Mass Distribution:

1. residual force offsets can be reduced by adjusting total body mass
2. residual moment offsets can be reduced by adjusting moment due to gravity by moving COM location