Simplification of the Hill Muscle Model Computation for Real-Time Walking Controllers with Large Time Steps

Nicolas Van der Noot*, Florin Dzeladini**, Auke J. Ijspeert** and Renaud Ronsse*
* Center for Research in Energy and Mechatronics, Université catholique de Louvain (UCL), Belgium
nicolas.vandernoot@uclouvain.be, renaud.ronsse@uclouvain.be
** Biorobotics Laboratory, École polytechnique fédérale de Lausanne (EPFL), Switzerland
florin.dzeladini@epfl.ch, auke.ijspeert@epfl.ch

1 Motivation

Bio-inspired controllers are emerging as a promising way to implement dynamic walking on humanoid robots without resorting to full local controllability concepts like the Zero-Moment Point-based methods [VB04]. Among all the bio-inspired approaches, we implemented the one proposed by Geyer and Herr [GH10], relying on reflex-controlled virtual Hill muscles. The forces generated by these muscles are determined by the length l_{ce} of the active, contractile element of each muscle-tendon unit (MTU). The update rate of l_{ce} is governed by the muscle-velocity relationship, as a stiff and strongly non-linear state equation. Consequently, this requires a small integration time step, which might lead to computational issues when transferring this model to real-time controllers. In this contribution, we illustrate that the dynamics induced by this muscle-velocity relationship is actually negligible for fast muscles. It can thus be replaced by a steady-state approximation. We compare three methods to compute this steady-state approximation, along with their impact on accuracy and computational cost. For slower muscles, we present a technique to mix both approaches: steady-state computation and full muscle dynamics-based models. The impact of the proposed simplification is evaluated in a forward simulation environment called Robotran [FS93], modelling a torque control robot, namely the COMAN [DMMC13], see Figure 1.

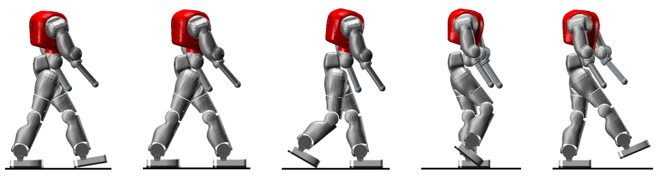

Figure 1: Walking gait of the COMAN in the Robotran simulator.

2 State of the Art and Proposed Solution

A first approach to get l_{ce} is to integrate the muscle-velocity relationship $\dot{l_{ce}}$ relative to time [GH10] depending on two given inputs: the total MTU length l_{mtu} and the level of activation A provided by the motor neuron:

$$\dot{l_{ce}} = f(l_{ce},\, l_{mtu},\, A) \tag{1}$$

However, the muscle-velocity relationship $f(\cdot)$ is so stiff and non-linear that it can cause huge oscillations of l_{ce} if the integration time step is too large, as illustrated in Figure 2. The critical time step duration strongly depends on the muscle properties. The reflexes of [GH10] directly use l_{ce} in the feedback loop, such that oscillations in l_{ce} propagate and would likely make the robot fall.

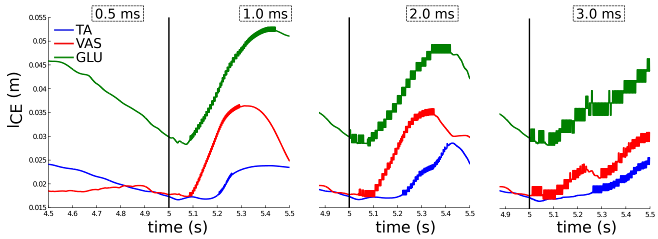

Figure 2: Temporal evolution of l_{ce} for three different muscles: tibialis anterior (TA), vastus muscle group (VAS) and gluteus muscle group (GLU). Initially, the integrator time step is set to 0.5 ms. At $t = 5s$, it is changed to 1, 2 or 3 ms.

So, avoiding this problem requires to keep the time step small enough or to use more advance integration schemes (we currently use a Euler explicit one). However, this is too greedy for some real-time controllers. Here, we propose to neglect the muscle dynamics and to consider that l_{ce} is always at steady-state, i.e. $\dot{l_{ce}} = f(\cdot) = 0$. We compare three methods to get this steady-state value in real-time.

1. A Look Up Table (LUT), generated off-line, stores the values of l_{ce} for many inputs values. Then, l_{ce} is interpolated for any inputs in real-time. While being quite efficient, this method accuracy depends on the inputs mesh refinement. If this refinement is too small, computational efficiency is deteriorated.

2. From this LUT, a third-order polynomial approximation was computed and used in real-time to compute the l_{ce} steady-state value. While this method is the most computationally efficient, its accuracy strongly depends on the LUT to fit, and so on the muscle properties.

3. A Newton-Raphson scheme was also tested to solve (1) at steady-state. Contrary to both previous methods, this one does not require a pre-process computation. Its main drawbacks are that it could converge to an unstable equilibrium point and that more than one iteration might be necessary to reach the desired accuracy. In the next section, results are reported only with a single iteration.

3 Current Results and discussion

Using similar muscle properties than [GH10], with proper dynamic scaling [SGT12], we compared the l_{ce} profiles provided by our three methods with the one of the original model. We observed that the two first methods performed better on proximal muscles than on distal ones. The polynomial method was more computationally efficient, but the LUT one was more accurate, at least if the input mesh was fine enough. The third method provided good accuracy on all muscles except the hip flexor (HFL). These results suggested that neglecting the muscle-velocity relationship dynamics has indeed a very limited impact on the l_{ce} profile. The three proposed methods and the one to select depend on the muscle properties and on the controller requirements.

The reason why the muscle-velocity dynamics can be neglected is because its time constant is actually very small compared to the controller time-step. We now consider the case of the muscle with the slowest dynamics: the soleus muscle (SOL), see Figure 3. The reference l_{ce} was computed with a 0.5 ms time-step (2000 Hz) while the proposed approximations were computed with a 10 ms time step (100 Hz). Using the full dynamics model [GH10] (green signal) results in the same problems as the ones presented in Figure 2. Using the third-order polynomial approximation (red signal) deviates from the actual reference for two reasons (i) the fit with the steady-state value is not perfect and (ii) more significantly, the dynamics of this muscle is too slow to be considered as negligible with respect to the sampling frequency of 100 Hz. Combining these two approaches (blue signal) actually provides the best approximation of the muscle dynamics, even at 100 Hz: the full dynamics model is used when $\dot{l_{ce}}$ is not too large, otherwise, the steady-state approximation is used.

Interestingly, directly replacing the full dynamics model [GH10] with any of the three steady-state approximation methods on the simulated COMAN preserved walking stability, although its gait became more jerky and less robust to perturbations, especially for the third-order polynomial approximation method. Re-optimizing the reflex rules led to retrieve more robust gaits. On top of that, these walking gaits coped with time steps up to 3 ms, while the full dynamics model required a maximal time step of 0.5 ms. Even if the muscle simplification still holds above 3 ms (see Figure 3), higher time steps caused issues in the controller reflex rules refreshment.

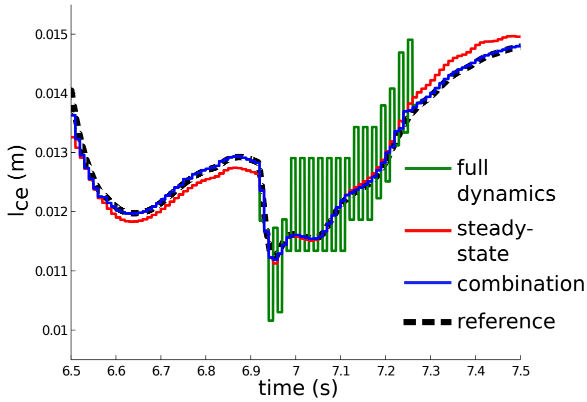

Figure 3: Temporal evolution of the SOL muscle l_{ce} for the COMAN walking with the full dynamics model and a Hill time step integration of 0.5 ms. The *reference* signal is the one used in the controller (computed with full dynamics and 0.5 ms time step). All other signals are computed with a 10 ms time step. The *full dynamics* signal integrates $\dot{l_{ce}}$, the *steady-state* one uses the third-order polynomial approximation and the *combination* signal is a mix between these two methods.

Acknowledgement

This work is supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under Grant 611832 (WALK-MAN), by UCLouvain (internationalization grant, IMMC11/13), and F.R.S.-FNRS (Crédit aux Chercheurs, 1.5025.12).

References

[VB04] M. Vukobratovic and B. Borovac. Zero-moment point - thirty five years of its life. *International Journal of Humanoid Robotics*, 1(1):157 – 173, 2004.

[GH10] H. Geyer and H. Herr. A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 18(3):263 – 273, 2010.

[FS93] P. Fisette and J.C. Samin. Robotran: Symbolic generation of multi-body system dynamic equations. In Werner Schiehlen, editor, *Advanced Multibody System Dynamics*, volume 20 of *Solid Mechanics and Its Applications*, pages 373–378. Springer Netherlands, 1993.

[DMMC13] H. Dallali, M. Mosadeghzad, G. Medrano-Cerda, N. Docquier, P. Kormushev, Tsagarakis N., Li Z., and Caldwell D. Development of a dynamic simulator for a compliant humanoid robot based on a symbolic multibody approach. *2013 IEEE International Conference on Mechatronics (ICM)*, pages 598–603, February 2013.

[SGT12] A. Schepelmann, H. Geyer, and M. Taylor. Development of a testbed for robotic neuromuscular controllers. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2012.