EG UK Computer Graphics & Visual Computing (2018)
G. Tam and F. Vidal (Editors)

Keys-to-Sim: Transferring Hand-Crafted Key-framed Animations to Simulated Figures using Wide Band Stochastic Trajectory Optimization
—
Supplementary Material

Dominik Borer[1,2] Martin Guay[1] Robert W. Sumner[1,2]

[1]Disney Research [2]ETH Zürich

Appendix A: Factored Linear Feedback

We trained a reduced (factored) linear function (as in [DLvdPY15]) that maps a deviation of the current character state $\mathbf{s}$ from the reference state $\hat{\mathbf{s}}$ to a deviation from the reference action $\hat{\mathbf{a}}$ to the final action $\mathbf{a}$, which can formally be written as:

$$\delta\mathbf{a} = \mathbf{M}_{feedback} \cdot \delta\mathbf{s},$$

where $\delta\mathbf{s} = \mathbf{s} - \hat{\mathbf{s}}$. The final action is then $\mathbf{a} = \hat{\mathbf{a}} + \delta\mathbf{a}$. Instead of optimizing for the full feedback matrix $\mathbf{M}_{feedback}$ of size $m \times n$, where m is the dimension of $\delta\mathbf{a}$ and n the dimension of $\delta\mathbf{s}$, the matrix is factored into two projection components $\mathbf{M}_{sp}$ and $\mathbf{M}_{ap}$ of dimensionality $r \times n$ and $m \times r$, resulting in:

$$\delta\mathbf{a} = \mathbf{M}_{cp} \cdot \mathbf{M}_{ip} \cdot \delta\mathbf{s}.$$

This factored policy with reduced-order r has only $r \cdot (m+n)$ parameters to optimize. For our experiments we used $r = 1$.

The goal is then to take an open-loop reference action and the corresponding simulated states (feasible reference motion), and optimize this factored linear feedback function such that the motion can be repeated multiple times without falling.

Reference State and Controls In our case, the reference controls are the PD joint angles obtained through the trajectory optimization and the reference states come from the corresponding simulated motion. The character state used is defined as a 19-dimensional state vector, comprised of a set of key-features describing the characters pose and motion:

$$\mathbf{s} = \{\mathbf{q}_{root}, h_{root}, \mathbf{M}_{root}, \mathbf{p}_{com}, \mathbf{v}_{com}, \mathbf{d}_{feet}\},$$

where $\mathbf{q}_{root}$, h_{root} and $\mathbf{M}_{root}$ are the orientation, height and angular momentum of the root, $\mathbf{p}_{com}$ and $\mathbf{v}_{com}$ the position and linear velocity of the center-of-mass (CoM) and $\mathbf{d}_{feet}$ the vectors from the CoM to the feet. All quantities are expressed in the character frame, which is defined as the root orientation around the vertical axis.

Optimization Details

We optimize the matrix parameters with CMA-ES. Because it is a difficult problem, it is performed in multiple stages, gradually increasing the difficulty by performing more rollouts of the motion, specifically we do 2, 4, 8, 16 and 32 rollouts. For each stage the initial mean of CMA-ES is initialized with the best solution of the previous stage (zero at the beginning). The initial standard deviation is set to $\sigma = 0.1$, the population size to 30 and CMA-ES is executed for 1000 iterations.

During the optimization the motion is simulated for the specified number of rollouts, using the sampled matrix parameters, and evaluated using cost functions (1) measuring how long the character was able to stay in balance and how close the simulated motion is to the reference motion. Similar to the trajectory optimization the cost is computed at a low frequency, every 0.05s. Although the reference motion is mechanically feasible (output from the trajectory optimization), it cannot be repeated multiple times and thus strictly measuring similarities between poses is not enough. Additionally balance and root features help to prevent the character from falling. Hence the final cost energy to minimize is:

$$E_{motion} = E_{root} + E_{balance} + E_{pose} \quad (1)$$

Root E_{root} tries to match the root height to the reference motion. It is the same as E_h defined for the trajectory optimization.

Balance $E_{balance}$ tries to keep the character in balance:

$$E_{balance} = w_{balance} \cdot (t_{total} - t_{balance}),$$

where t_{total} is the total time of the repeated reference motion for the current stage (reference motion duration $\times$ number of rollouts) and $t_{balance}$ is the time the character was able to stay in balance, which is defined to have the root height above a certain threshold (we used half of the initial root height). We used $w_{balanec} = 200$.

Pose E_{pose} tries to match the internal joint angles to the reference motion. It is the same as defined for the trajectory optimization.

References

[DLvdPY15] DING K., LIU L., VAN DE PANNE M., YIN K.: Learning reduced-order feedback policies for motion skills. In *Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (2015), ACM, pp. 83–92.