Computational climbing for physics-based characters

Tim Gorm Kaas-Rasmussen Olsen[1,2], Sheldon Andrews[1], and Paul G. Kry[1]
McGill University[1] University of Copenhagen[2]

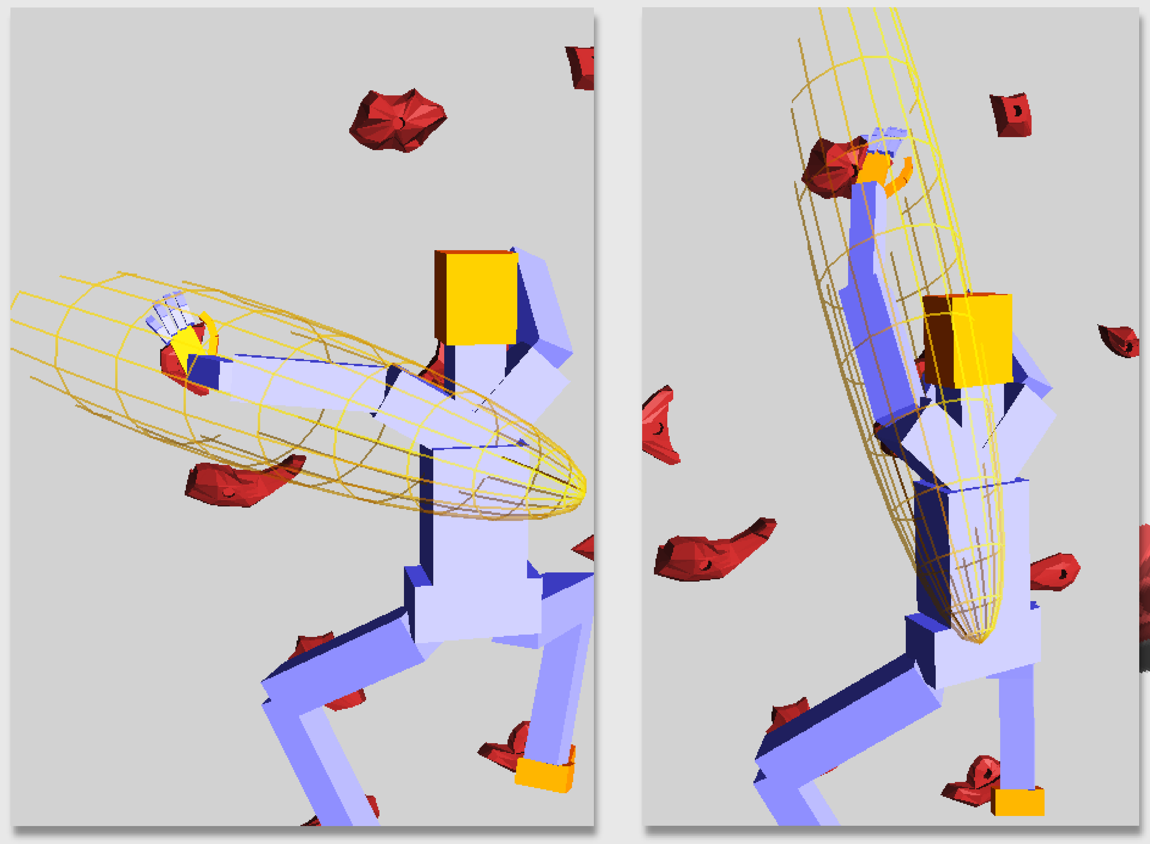
Introduction

We present our ongoing work to create realistic climbing animations for articulated 3D characters. Our goal is to automatically synthesize physically plausible motions for highly detailed models. Specifically, for characters with attached hands, giving them true grasping capabilities. Our method chooses limb configurations that are suitable for climbing, favoring postures with the ability to effectively exchange forces between the character and its environment. Grasping postures are learned that support the character as it climbs in a dynamic environment. And a novel linear feedback controller is used to adjust the grip strength in order to deal with force variations at the wrist.

Character Model

The physics-based character model has 83 degrees of freedom (DOF), including 21 DOF for each hand, and a mass distribution similar to that of a real human. The RTQL8 physics engine is used for real-time simulation.

Motion Planning



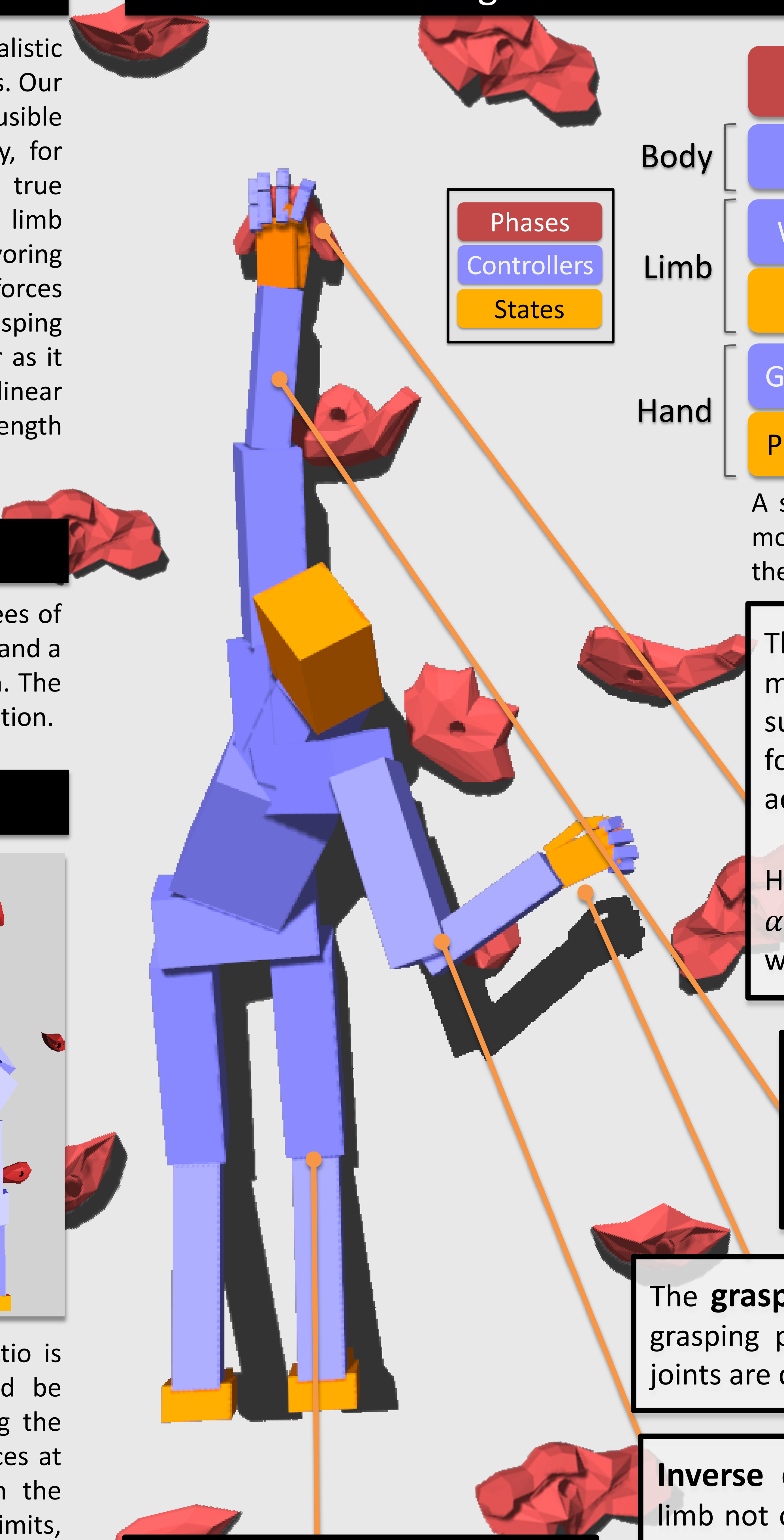
A heuristic based on the force transmission ratio is used to determine whether any limbs should be repositioned. The FTR is calculated by mapping the potential joint torques to a set of potential forces at the end effector, visualized by an ellipsoid in the above figure. If joints of an active limb are near limits, or if the FTR in the climbing direction drops below a certain threshold, a new target handhold is selected for that limb. A kinematic sampling process based on the method of Tonneau et al. [1] is used to select a reachable handhold with a skeleton configuration that best allows the character to push / pull in the required direction with the limb being retargeted.

Controllers & Strategies

Legend: Phases / Controllers / States

	Reach for handhold	Grasping	Advance climbing
Body	Body SPD – Non-ragdoll effects, track kinematic pose		
Limb	Wrist Inverse Dynamics	Limb $J^T F$	Motion planning
Limb	Wrist track spline	Pull/push to advance	
Hand	Grasp SPD	Grasp SPD / Finger $J^T F$	
Hand	Pre-shape	Close hand / Adjust Grip	

A state machine is used to activate controllers depending on the phase of motion and state of the simulation. Various strategies are used to accomplish the climbing task.



The **grip controller** provides online adjustments to the grasp by modulating contact forces at each finger. The force needed to support the character f_s is estimated from inertial and external forces. A linear feedback rule is used to compute the force adjustments at each finger segment:

$$f_i = \alpha\, \hat{n}\, \hat{n}^T f_s\,.$$

Here, $\hat{n}$ is the unit length contact normal at the finger segment and α is a gain parameter. Joint torques are computed by $\tau = J_i^T f_i$, where J_i is the Jacobian of finger segment i at the contact point.

A **virtual force** is generated at each supporting limb according to the desired climbing direction $\vec{v}$. Limb joint torques are computed by $\tau = J_l^T \vec{v}$, where J_l is the linear Jacobian of the end effector for limb l.

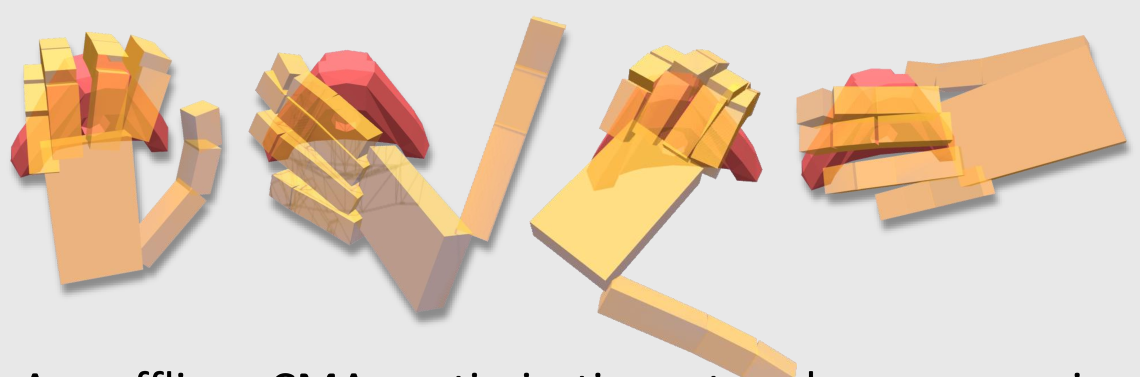
The **grasp controller** pre-shapes and closes the hand according to a grasping posture learned by the optimization step. Torques at finger joints are computed using stable proportional derivative (SPD) servos [2].

Inverse dynamics is used to re-target a limb not currently supporting the character. The end effector body (palm) is maneuvered to an orientation and position for pre-shaping and grasping the target handhold. The trajectory is stored as a Hermite spline.

A **pose tracking** controller maintains a natural configuration for joints not actively used for climbing. SPD servos [2] are used here too.

Example videos
goo.gl/HNrrcz

Grasp Optimization



An offline CMA optimization step learns grasping postures that maximize the forces exchanged due to contacts between the fingers and a candidate handhold. A database of grasping postures is learned for each handhold, and for various pulling directions. Rather than considering all 21 DOF of the hand independently, the optimization uses a synergy of hand postures captured from real climbing tasks by a *ShapeHand* sensor. This lowers the dimensionality of the search to 3 DOF for the reduced pose basis and 6 DOF for the position and orientation of the palm.

Discussion & Future Work

The preliminary results shown here are encouraging. we are confident that realistic climbing animations for high fidelity character models may be synthesized with our approach. Aside from an offline optimization step, little computational overhead is required, making it suitable for games and other real-time applications. Currently, we are investigating methods to perform online grasp planning, eliminating the need for a costly non-linear optimization step. We hope to make climbing a ubiquitous activity available to characters in any physics-based virtual environment. Furthermore, data collected with the sensor augmented climbing wall, shown in the figure on the left, will be used to determine other climbing strategies.

References

[1] Tonneau S., Pettré J., Multon F.: Task Efficient Contact Configurations for Arbitrary Virtual Creatures. In *Proceedings of Graphics Interface 2014, pp. 9-16.*
[2] Tan J., Liu C. K., Turk G.: Stable Proportional-Derivative Controllers. *IEEE Computer Graphics and Applications*, 31(4):34–44, July 2011.