Learning to walk using genetic algorithms

Dennis Struhs

Arbeitsbereich TAMS

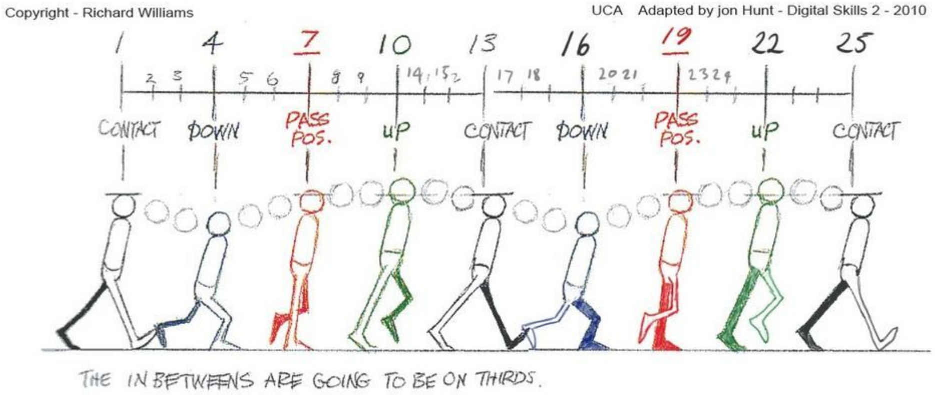
12.11.2018

Table of contents

Introduction

The art of walking

For us humans the act of walking is trivial, whereas for a Robot it is not. The question of interest here is: What characterizes our way of walking and how can we make robots walk the same way as we do?



Source: `https://www.schoolofmotion.com/blog/walk-cycle-inspiration`

Manually programmed walking

Every servo motion is explicitly specified and controlled by the programmer, including the sensory data in order to keep the robot in balance.

Pros:

- Fairly straight forward to do.
- The code is easier to understand.

Cons:

- Produces usually very crude motions.
- Does not adapt well to sudden environment changes.
- Impracticable for complicated motions.

Room for improvement

A better approach for creating walk cycles for robots without having to manually specify every motor motion is required. A better Idea would be to use:

- Sensory feedback.
- Mathematical models.
- Feedback loops.

In order to specify generic algorithms and control structures that completely handle the walk cycle generation without much intervention of the programmer.

Using Algorithms

Recent approaches use the ZMP (Zero moment point)[2] approach to produce stable walk cycles. The basic Idea behind it is that the point, which is the sum of gravitational and acceleration force, always remains inside the supported range. This can be mathematically expressed as:

$$\boxed{T_{ZMP} = T_{Com} + T''_{Com}} \tag{1}$$

with T_{Com} being the relative position of the center of gravity and T''_{Com} the acceleration of the center of gravity.

Limitations

- Requires tweaking of values if something on the robot body layout changes.
- Relies on the model, specified by the user, to be correct.
- Limited reaction to environment changes or perturbations upon the robot itself.

How can we improve?

- We're looking for an approach that works ideally for any kind of multipedal robot.
- Which automatically adjusts for robot body changes.
- Offers maximum robustness to environment changes and perturbations

A different approach

So far we've been either:

- Creating the full walk cycle completely manually.
- Creating the walking motion by using a world model specified by the user, petrified into an algorithm that handles the walk cycle using feedback from sensors.

However if we look back at how we humans learn to walk, being through trial and error, another option to create a walking cycle for a robot pops up: **Genetic algorithms**.

Genetic algorithms

Genetic algorithms: Overview

Are an computational problem solving method[5], characterized by:

- Being generate-and-test algorithms.
- Being population based.
- Using stochastic methods.
- Using search procedures.
    - Guided by fitness.
    - Driven by variation and selection operators.
- Being anytime algorithms which means that they can stop at any given time with a (suboptimal) solution.

Genetic algorithms: Visualization

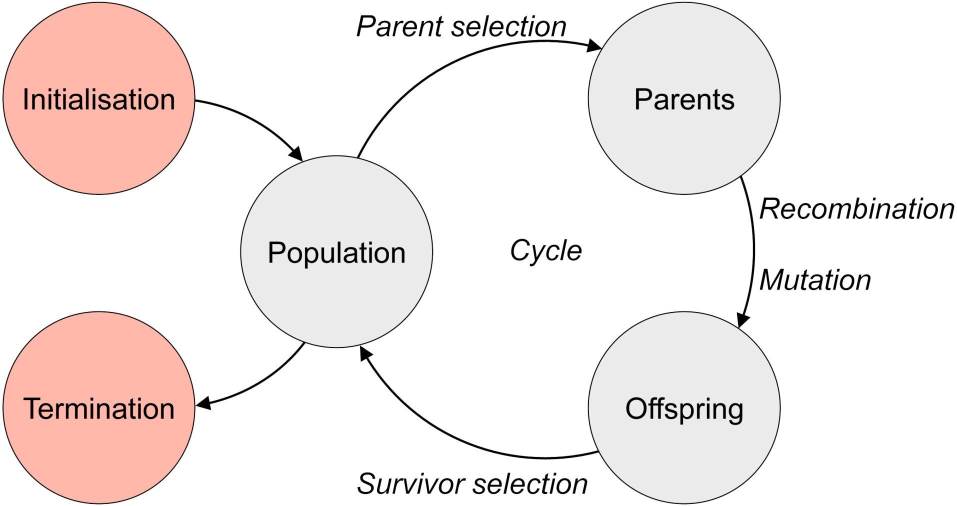


Figure: Basic execution scheme of genetic algorithms. Source:[5]

Genetic algorithms: Fitness function

Represents requirements that the algorithm should adapt to. It is used to select the parents for the next iteration and assigns a measure for quality (fitness) to the genotypes.

For example:

- **Context:** Minimize x^2.
- **Phenotype:** $x \in \mathbb{N}$.
- **Genotype:** z: binary representation of x.
- **Fitness Function:** fitness $f(z)$ of genotype z is defined as 1 divided by the square of its corresponding phenotype:

$$z = 0010_2 \rightarrow \text{phenotype} : x = 2 \rightarrow f(z) = \frac{1}{x^2} = 0.25 \tag{2}$$

The fitness landscape

A model that represents the fitness value for each possible trait combination.

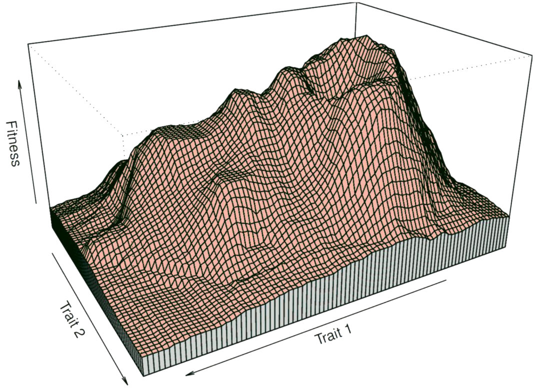


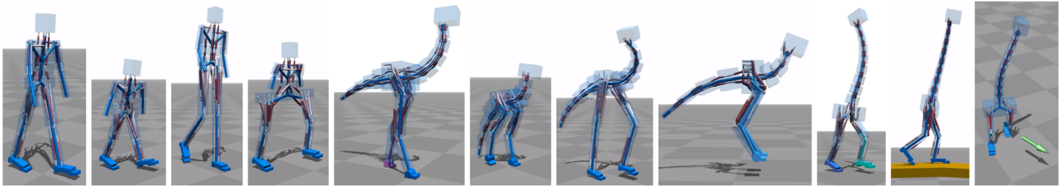
Figure: Example plot of some fitness landscape.Source:[5]

- For common applications of a very **high dimension**.
- Local optimum: better solution than neighboring solutions.

  Which divides further into:
  - Uni-modal problem: only one local in the landscape.
  - Multi-modal problem: several local optima in the landscape.
- Global optimum: the best possible solution over the entire landscape.

Genetic walking

A new approach: Virtual muscles



Source: [1]

The approach as suggested by [1] uses genetic algorithms for simulated bipeds in which both the muscle routing and control parameters are optimized.

The muscle model

The specific muscle model here as suggested by [1] is the Hill-Type muscle which consists mainly of three elements:

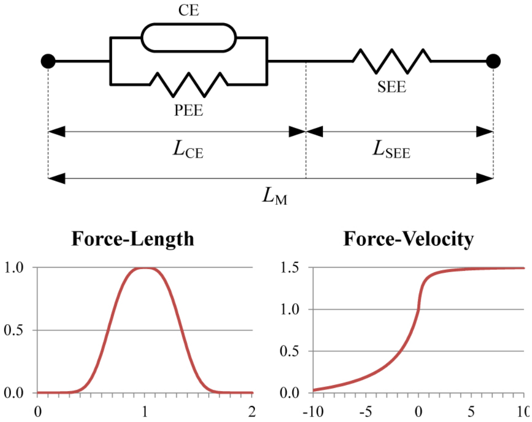
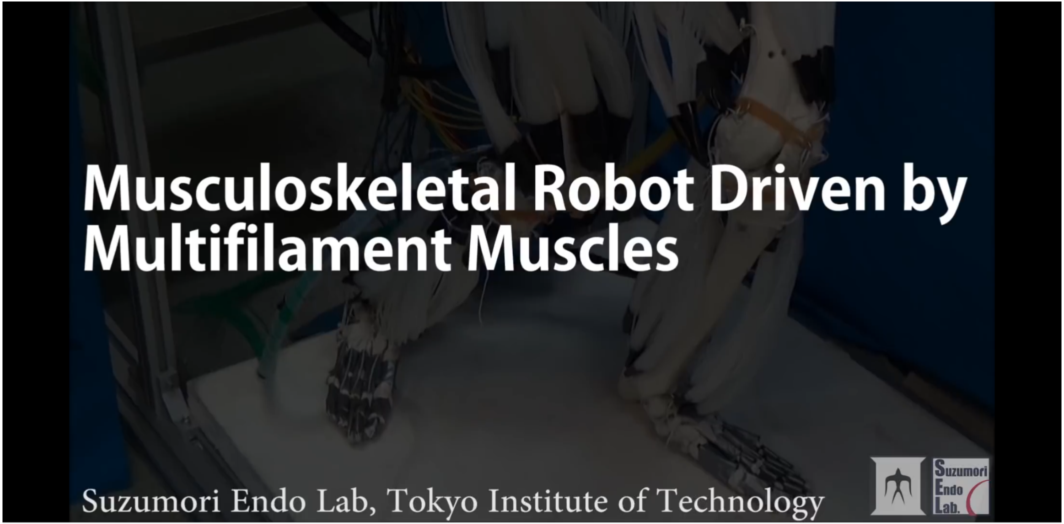


Figure: Hill-Type muscle model. Source: [1]

- *A contractile element (CE)* which represents the contracting muscle fibers based on the muscle activation state.
- *A parallel elastic element (PEE)* that represents the passive elastic material surrounding muscle fibers.
- *A serial elastic element (SEE)* representing the tendons that connect the muscle to the bones.

The relevance for robotics



Source: `https://www.youtube.com/watch?v=0ZBD2tcK0U4`

The optimization

The routing of any muscle M is defined through a vector of n attachment points $[\{b_1, p_1, \} \ldots, \{b_n, p_n\}]$:

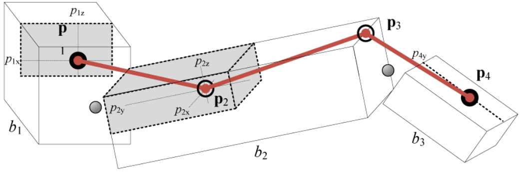


Figure: The muscle attachment points that will be optimized within a constrained region. Source: [1]

- Is defined as an attachment point p_i and a body b_i to which the point is attached to.
- Each point p_i is a fixed offset in the coordinate frame of body b_i; it moves along with the body it is attached to.
- Multiple points can be attached to a single body.
- p_1 and p_n, represent the locations where the muscle tendons are attached to the skeleton.

Controlling the muscles

At each step the finite state machine (FSM) for each leg is updated, based on the current leg state.

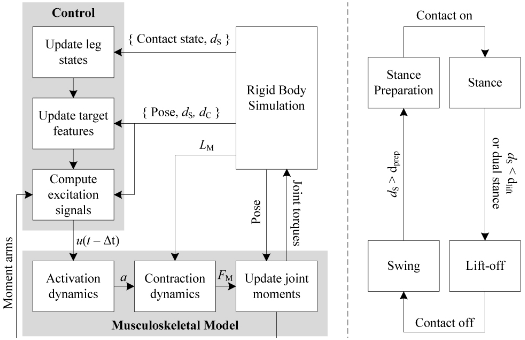


Figure: The muscle attachment points that will be optimized within a constrained region. Source: [1]

The current leg state can assume one of the following states:

1. Stance
2. Lift-off
3. Swing
4. Stance preparation

State transitions occur after ground contact if the reaction force is below the threshold value $F_{Contact}$

Target features

An important part of the control system is based on so called target poses, which are defined for a set of featured body parts.

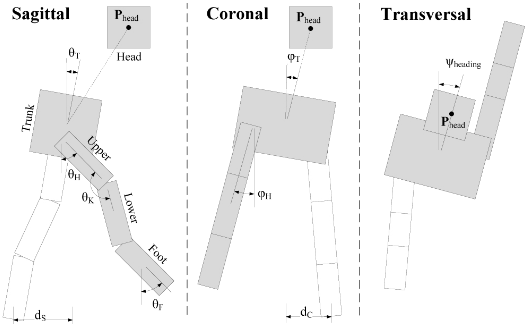


Figure: Target features for trunk, head and swing leg, shown in sagittal, coronal and transversal projection.Source:[1]

More specifically,
there are defined target orientations for:

- A trunk body.
- A target position.
- A head body.
- The leg segments.

Muscle based feature control

Is performed in order to find the muscle excitation levels that cause any relevant muscle M to drive the featured bodies towards their respective targets.

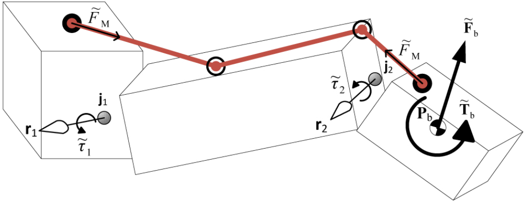
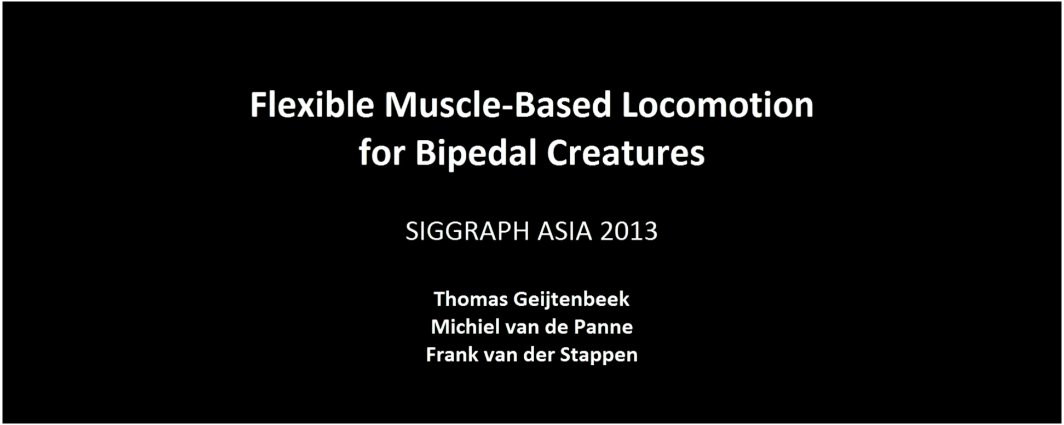


Figure: The muscle attachment points that will be optimized within a constrained region.Source:[1]

- Uses a muscle-based variation of the Jacobian transpose [3] control.
- It finds a set of muscle torques that emulates the effect of a virtual force or torque, see figure 7.

Experiments



Source: `https://www.youtube.com/watch?v=pgaEE27nsQw`

Conclusion

Summary

There was a flexible framework introduced for muscle-based motion of bipedal creatures. The method can support:

- Various variations of creatures.
- A range of possible speeds.
- Turning behavior and robustness to external disturbances.
- Unexpected changes in terrain slope.

The key features revolve around the optimization of muscle routing and the usage of muscle-based approximations to the Jacobian transpose control. This allows for flexible and robust full 3D muscle driven motion for various bipedal creatures.

Limitations

The current approach however still has limitations. Compared to [4] the humanoid walking and running motions are of somewhat lesser quality, considering especially the upper body movement.

This can be partially explained by the lack of upper body target features on the humanoid creatures. These targets were omitted in favor of a more generic approach, yet by focusing on a more elaborate humanoid model, researches can easily reintroduce these and more domain-specific elements.

Questions? Dennis Struhs E-Mail: 3struhs@informatik.uni-hamburg.de

Literature I

- Thomas Geijtenbeek, Michiel van de Panne, and A. Frank van der Stappen.
  Flexible Muscle-Based Locomotion for Bipedal Creatures - https://dl.acm.org/citation.cfm?id=2508399&dl=ACM&coll=DL.
  (1):11, 2013.

- Robert Schmidt.
  Development of a stable robot walking algorithm using center-of-gravity control - https://robocup.informatik.uni-hamburg.de/wp-content/uploads/2015/05/thesis-robert.pdf.
  (1):46, 2015.

Literature II

Craig Sunada, Dalila Argaez, Steven Dubowsky, and Constantinos Mavroidis.
A Coordinated Jacobian Transpose Control for Mobile Multi-limbed Robotic Systems - https://pdfs.semanticscholar.org/ac97/376734be3782560fa7bfeadb36068b0cb682.pdf.
Department of Mechanical Engineering, Massachusetts Institute of Technology Cambridge, MA, 02139, (1):11, 1994.

J. Wang, S. Hamner, S. Delp, and V. Koltun.
Development of a stable robot walking algorithm using center-of-gravity control - https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4523558/pdf/nihms641752.pdf.
31, (4):25, 2012.

Literature III

Prof. Stefan Wermter.
Biological inspired AI lecture -
https://www.inf.uni-hamburg.de/en/inst/ab/wtm/teaching/teaching-2018-wise-bioinspired-ai-lecture.html, 2017.