



**HAL**  
open science

## Physics-based control of walking virtual characters in low frequency simulations

Samuel Carensac, Nicolas Pronost, Saïda Bouakaz

► **To cite this version:**

Samuel Carensac, Nicolas Pronost, Saïda Bouakaz. Physics-based control of walking virtual characters in low frequency simulations. 31st International Conference on Computer Animation and Social Agents CASA 2018, May 2018, Pekin, China. pp.77-82, 10.1145/3205326.3205361 . hal-01884827

**HAL Id: hal-01884827**

**<https://hal.science/hal-01884827>**

Submitted on 19 Oct 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Physics-based control of walking virtual characters in low frequency simulations

Samuel Carensac  
INSA de Lyon  
Univ Lyon, LIRIS CNRS UMR5205  
samuel.carensac@insa-lyon.fr

Nicolas Pronost  
Université Lyon 1  
Univ Lyon, LIRIS CNRS UMR5205  
nicolas.pronost@univ-lyon1.fr

Saïda Bouakaz  
Université Lyon 1  
Univ Lyon, LIRIS CNRS UMR5205  
saida.bouakaz@univ-lyon1.fr

## ABSTRACT

Physics-based control of virtual characters traditionally uses high simulation frequencies of 1 to 2 kHz. While lowering the simulation frequency frees computation time, it usually introduces instabilities within the simulation. In this paper, we propose a control strategy that can be used for high and low simulation frequencies, down to 225 Hz. The inherent instabilities were reduced by optimizing control parameters and by introducing a novel control feedback for the stance leg. We also show how lower frequencies hold a more restrictive space of possible control parameters than higher ones. Our controller shows equal robustness as high frequency controllers while requiring in average only 0.8 ms per simulation step.

## CCS CONCEPTS

• **Computing methodologies** → **Physical simulation**; *Animation*;

## KEYWORDS

motion control, physics-based animation, virtual character, simulation frequency

### ACM Reference Format:

Samuel Carensac, Nicolas Pronost, and Saïda Bouakaz. 2018. Physics-based control of walking virtual characters in low frequency simulations. In *CASA 2018: 31st International Conference on Computer Animation and Social Agents, May 21–23, 2018, Beijing, China*. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3205326.3205361>

## 1 INTRODUCTION

In recent years, the physical simulation of human locomotion has aroused a great deal of interest among researchers in bio-mechanics, robotics and computer science. Indeed, physical simulation provides an elegant and appropriate method for generating physically realistic interactions between a character and his environment. Even though many control strategies allowing a real-time simulation have been proposed [3], very few could be used in a virtual environment containing a sizable number of physically simulated objects in addition to the controlled character. One of the reasons is that simple control strategies require a high simulation frequency (1 kHz or more). Since the cost of simulating various elements of the environment does not depend on the frequency, high simulation frequencies limit the use of those controllers to simple environments. Recent works propose controllers for lower simulation frequencies but require complex control strategies to compensate for the larger

time interval between each control step. These systems require large computation times to control a character thereby limiting the benefits of a low simulation frequency. To our knowledge, the study of the effect of low frequencies on simple control strategies has not yet been investigated.

In this paper, we present a physics-based motion controller that can be used for high and low simulation frequencies. In order to produce motions at least as robust, controllable and reactive as the ones obtained with a high frequency, we propose novel components such as a ground contact stabilizer and a heading management. To fully benefit from low frequencies, we paid a special attention to keep a low computation cost for each simulation step.

The main contributions are twofold. We show that by carefully calculating the values of some control parameters, high-frequency controllers can be used at much lower frequencies than the ones required in previous works without any further modification to the controller. We propose a process to find these values across a large interval of frequencies. Finally, we propose a novel online optimization component that reduces instabilities induced by lowering the simulation frequency.

The reminder of this paper is organized as follows: section 2 reviews previous works. Section 3 describes our low frequency controller and online optimization. Section 4 presents our method to evaluate the control parameters. We present our results and discuss the method in section 5. In section 6, we summarize our approach and highlight future works.

## 2 PREVIOUS WORKS

Although using lower frequencies frees large amounts of computation time, it also reduces the stability of the physics simulation, especially with respect to rigid bodies collisions. Recently, some works have been able to design motion controllers using low control frequencies. Muico et al.'s [12, 13] use a prediction of the contact forces and a nonlinear quadratic regulator (NQR) to track a reference trajectory to allow the control of a character at 120 Hz. Although this system achieves real-time control, the average computation time for one simulation step with numerous contacts with the ground remains around 6 ms. Model predictive control (MPC) has recently been used to create controllers extremely robust to external perturbations. However, its computational complexity restricts its real-time application [15]. Han et al.'s controller [6] produces real-time simulations of a 3D model at 60 Hz by combining a MPC with a guiding reference motion but the available resources are completely used for high energy motions. Liu et al. [10] proposed a sampling-based controller (SAMCON) that is very stable but requires an extremely high quantity of computational resources to find the correct samples. Greer [5] modified the SAMCON for cyclic

motions to be able to reuse samples between cycles. This method allows them to create a 60 Hz controller and using only an average of 0.62 ms per simulation step. However, this system is limited to cyclic motor tasks and, to be able to reuse the samples, the poses occupied by the character must not vary between successive cycles. In [9] a SAMCON controller is used to pre-generate samples and creates a control graph from prerecorded motion clips. The resulting controller runs at 200 Hz, and ten times faster than real-time, but it requires pre-computation of the samples and available clips.

Using low simulation frequencies can lead to numerical instabilities in the physics simulation, and in virtual character control, especially in the reaction forces at ground contact. Several methods have been proposed to reduce such instabilities. We can sort them into two categories. In the first category, the physics engine itself is modified. For instance, Han et al. [6] chose to use the smoothed contact dynamics proposed in [16] resulting in more stable contacts. Unfortunately this system is not yet implemented in commonly used engines such as ODE or PhysX. Liu et al. [9] uses a modified version of ODE proposed in [11] that implements an implicit damping scheme to obtain stabler contacts. The use of soft models for the foot [8] can also be used to reduce the instabilities at the cost of computation time required for soft contacts. In the second category, dedicated components are added on the controller side to limit the impact of the numerical instabilities. Muico et al. [12] integrated a prediction of the contact forces into the controller to compute torques compensating for the instabilities. The main drawback of this solution is that it needs knowledge on the inner working of the engine which limits its use to open source engines.

The SIMBICON model and associated works [1, 2, 17] propose a high-level controllable system (character’s orientation, velocity and step positions). Those systems typically run at 2 kHz and they can still easily allow real-time interactive simulations thanks to their very low computation time. Few works tried to study the impact of lowering the control frequencies in such system. Giovanni et al.’s [4] used the basic model in various engines and showed that below a frequency of 750 Hz the system becomes unstable no matter the engine. Greer [5] implemented various improvements such as angular momentum regulation or fixing of the stance foot but they did not obtain successful results at low frequencies.

In this article, we focus on the design of a control strategy that can be used for high and low simulation frequencies (2 kHz to 225 Hz). The proposed method is simple enough to keep a low computational time, but is sufficiently robust to ensure stability of motions.

### 3 CONTROL FRAMEWORK

Our system is built on the version of SIMBICON presented in [2]. This system uses Yin’s version [17] with key poses as input data for each joint except the swing leg joints. The target pose for the swing leg joints are computed by an inverted pendulum model (IPM). It also uses components that directly compute additional torques. One component is a gravity compensator that applies torques compensating for the weight of the limbs for all joints but the stance ankle [2, 5].

The second component controls the velocity of the character by applying a virtual force to the center of mass of the character. We use

a more recent version of this component exhibiting better stability by considering an adaptive target velocity within a walking step [1]. Their approach uses a curve storing the required variations of the desired velocity and an offset on this curve that gives control over the average velocity of the center of mass of the character. These two components are dynamically adapted at the end of each character step depending on the velocity that was observed during the step. We modified the learning process of the curve by normalizing the observed velocity so that the median value is the same as the median value of the desired velocity. This modification allows for a faster convergence by offering a better separation of the information contained by the desired velocity curve and the offset.

We also removed the linear balance feedback controllers to reduce the number of input parameters necessary for the simulation. This has the advantage of reducing the possible causes for instabilities when using the controller across various simulation frequencies. The controller is still robust to perturbations due to the IPM and the improved velocity controller.

Finally, we modified the way desired heading modifications are handled. Instead of applying the new desired heading instantaneously, we distribute it over two character steps. During each of those steps, we slowly transition from the old desired heading to the new one over a period of 0.3s by using linearly interpolated intermediate values. Doing the transition over two character steps will look natural since it is done over a single support phase, i.e. only one foot strike has happened between the user’s change of heading and the moment we reach the desired heading.

The model of the character is the same as in [17] (28 DOFs) except that we use a cuboid to represent the toes instead of a sphere.

#### 3.1 Ground contact stabilizer

The goal of the ground contact stabilizer is to compensate for the instabilities observed when the stance foot is in contact with the ground. In an ideal situation of full contact for a foot modeled with a cuboid, the four lower corners should be in contact with the ground. In practice, it is common to observe only three or less corners in contact. Also it is possible to observe sudden variations of which corners are in contact between successive simulation steps. Having a low number of contact points can lead to rotations of the stance leg and therefore unbalance the character often to its fall. Without enough contact points the friction forces applied on the foot are not high enough to cancel out the rotation provoked by the torque applied on the stance hip to control the orientation of the pelvis. In our approach we propose to design a component, the stabilizer, to ensure at least three contact points, and thus highly limiting the possibility of any undesired rotation. Our approach is inspired by sampling-based solutions using an online optimization as they are able to find pose displacements resulting in stable contacts even with extremely low frequency simulations [5]. We choose to not use a prevision window and handle each simulation step independently trading stability for a lower computational cost.

*3.1.1 Pose samples.* In [5], the samples define a target pose displacement while in our approach the samples define a supplementary torque to apply on the stance leg joints. This makes our system independent from input data, in particular independent from the PD-controllers gains. In our approach, we also limit the

samples to the joints the most directly involved in the contact with the ground in order to reduce the search space of the sampling algorithm. We selected the joints located in the stance leg as it is highly improbable that small variations of the torque applied on the other joints will have any significant impact on the contacts. We chose to exclude the toes joint as it only has an impact on the distribution of the forces between the toes and the foot and not on the total forces seen by the character. We excluded also the hip joint since in our system it is used to control the pelvis and not the leg. Our samples are then simply made of the ankle joint (2 DOFs) and knee joint (1 DOF) of the stance leg, resulting in a 3-dimensional search space.

**3.1.2 Simplified character model.** In addition to limiting our samples to stance leg joints we use a simplified character model during the evaluation of the samples allowing us to reduce its cost without introducing significant errors. In this model, we represent the removed body parts by a single force representing their weights. To ensure proper contact estimation, the stance leg is kept intact. We experimented with various combinations of removed body parts and studied their impact on the controller (see section 5.1).

**3.1.3 Samples evaluation.** Our evaluation of the contacts is done by evaluating the ground reaction forces (GRF) distribution in the stance foot. We will consider only their normal component to the ground and therefore will not consider the components resulting from the friction as the normal components are more stable and more reliable and the maximum tangential forces depend on the normal components (cone of friction).

Our goal is to ensure that a significant portion of the GRF is applied on each side of the foot, not to obtain a perfectly even distribution of the GRF. Our goal is similar to preventing the centroid of the GRF to get too close to a border of the foot. In practice, we try to find solutions keeping the centroid inside an area 15% smaller than the bottom side of foot.

We define the ratios  $r_{left/right} = F_{left/right}/F_{all}$  and  $r_{front/back} = F_{front/back}/F_{all}$  where  $F_{right/left}$  and  $F_{front/back}$  are the sums of the GRF on each respective side of the foot cuboid and  $F_{all}$  is the total sum of the GRF.

Our evaluation function is as follows:

$$f_{eval} = f_{quality} * 10 + f_{distance} \quad (1)$$

$$f_{quality} = \begin{cases} 1.0E15/10^{F_{all}/10+1} & F_{all} < 100 \\ (\max(l_{cor} - r_{left}, 0))^2 + \\ (\max(l_{cor} - r_{right}, 0))^2 + \\ (\max(l_{sag} - r_{front}, 0))^2 + \\ (\max(l_{sag} - r_{back}, 0))^2 & otherwise \end{cases} \quad (2)$$

$$f_{distance} = \max(\max(l_{cor}, r_{left\_init}) - r_{left}, 0)^2 + \\ \max(\max(l_{cor}, r_{right\_init}) - r_{right}, 0)^2 + \\ \max(\max(l_{sag}, r_{front\_init}) - r_{front}, 0)^2 + \\ \max(\max(l_{sag}, r_{back\_init}) - r_{back}, 0)^2 \quad (3)$$

Distribution of the GRF:  $f_{quality}$  represents the quality of the GRF distribution.  $l_{sag}$  and  $l_{cor}$  are user parameters controlling the level of restriction imposed on the distribution, respectively the

sagittal and coronal limits. Lower values lead to faster convergence but higher values lead to more stable contacts. In our experiments, we use a value of 0.15 for both parameters.

**Distance:** the goal of the function  $f_{distance}$  is to evaluate the distance between the GRF distribution without optimization ( $r_{cor\_init}$  and  $r_{sag\_init}$ ) and the GRF distribution using the evaluated sample. Our intention is to promote samples resembling the initial simulation and at the same time we limit  $r_{left/right\_init}$  and  $r_{front/back\_init}$  to respectively  $l_{sag}$  and  $l_{cor}$  to prevent the system from favoring samples with lower GRF distribution quality.

These two functions have a similar order of magnitude so we weight  $f_{quality}$  by a factor of 10 as the goal of  $f_{distance}$  is only to differentiate between already acceptable distributions (Eq. 1). We used the Covariance Matrix Adaptation (CMA) algorithm [7] to generate the samples. Any sample with  $f_{eval} < 1$  is accepted and the CMA is stopped, which is usually achieved in a couple of iterations.

## 4 GAINS STUDY

In physics-based animation of virtual characters, the gains (usually called  $K_p$  and  $K_d$ ) of the PD controllers used in the control feedback are commonly obtained from offline optimization. The cost function usually evaluates the quality and stability of the motion without constraining the gains [5]. Such optimization does not promote a specific set of gains as long as it produces an acceptable motion. Unfortunately, if the range of acceptable gains is small, the optimization process may fail to find a solution while one exists. In this section we study how the acceptable range of gains varies with the simulation frequency.

Intuitively we can imagine that acceptable gains for high frequencies may not be acceptable for lower frequencies. Indeed, when using lower frequencies the tracking of the desired pose will most likely be less precise as we correct the pose and velocity errors less often. Using constant gains, the larger pose and velocity errors will result in larger torques which will introduce vibrations and lead to a failure of the controller or the physics simulator. This observation makes us formulate the following hypothesis: lower frequencies have more restrictive ranges for acceptable gains. Following this idea, we used an offline optimization to find the range of gains (minimal and maximal values) for which the resulting motion is still similar to the one obtained with the original high frequency (1 kHz) and gains. The key poses are kept identical for all frequencies.

### 4.1 Objective function

The optimization process uses the following objective function:

$$f_{eval} = f_{gains} + \frac{\sum_{t < k} (f_{head} + f_{hands})}{k * frequency} + \\ f_{speed} + f_{balance} \quad (4)$$

where  $k$  is the duration of the evaluation in seconds and  $frequency$  the simulation frequency. This function is made of three parts. The first one numerically evaluates the set of gains:

- *evaluation of gains ( $f_{gains}$ ).* We define here if we want to find the minimal or the maximal gains using their sum. Each gain is normalized by its initial value to prevent the algorithm from favoring joints.

The second part of the objective function ensures that the new motion is similar to the original one. For that, we measure the displacements of end effectors, namely the head and the hands:

- *evaluation of hands* ( $f_{hands}$ ). To evaluate the quality of the hands positions, we used inverse kinematics to produce key poses giving a constant target position relative to the pelvis allowing us to evaluate any deviation from the target.
- *evaluation of head* ( $f_{head}$ ). This function penalizes a solution in which the head position does not stay in an elevation close to the one observed in the original frequency. We perform the same penalization for the neck rotation (as it has an impact on the head position) and the relative position of the head to the pelvis.

The last part contains two terms ensuring that global characteristics are conserved.  $f_{speed}$  discards any solution resulting in an error of more than 5% on the desired character speed and  $f_{balance}$  verifies that the character does not fall by checking the position of the center of mass.

## 4.2 Optimization strategy

As already mentioned, we do not directly use the sum of the gains but the sum of the gains normalized by their initial values to prevent the algorithm from favoring joints. To select correct initial values for each frequency, we made two assumptions: (1) the best gains for a frequency is close to the best gains for a slightly different frequency (continuity), and (2) lower frequencies have more restrictive acceptable ranges. With these assumptions we are likely to be able to use the solution obtained for a frequency as starting point for a slightly lower frequency.

The optimization procedure is given in Algorithm 1. The function `run_cma` runs an offline optimization using the CMA algorithm. Its first parameter is the input gains, the second is the simulation frequency and the last is a Boolean value specifying if we are looking for the minimal or the maximal gains. It executes the CMA-ES algorithm and stops either if it reaches 150 iterations or if there is no further evolution of the gains. We stop after 150 iterations so that we can update the normalization values often enough to prevent favoring the joints becoming significantly larger. The function returns the best set of gains and a Boolean set to false if no iteration of the CMA resulted in a stable motion.

## 5 RESULTS

Our controller was implemented using the ODE physics engine. The following performance are obtained on a single threaded implementation on an Intel(R) Xeon(R) CPU 3.0GHz. The simulation frequency is 300 Hz unless specified otherwise. Results on high-level control (response to perturbations, sharp turns ...) are presented in the companion video. All simulations use the same key poses and each frequency has a specific set of gains.

### 5.1 Ground contact stabilizer

In this section we report the effects of using various reduced models on computation time and the quality of the contacts between the feet and the ground. The following results have been obtained over simulations of 200 character steps of straight forward walking gait.

---

#### Algorithm 1 Offline optimization procedure

---

**Require:** a set of gains  $S_{ini}$  resulting in a stable motion and the associated frequency  $F_{ini}$

```

1: procedure LOWERFREQUENCY( $S_{ini}, F_{ini}$ )
2:   ( $S_i, F_i$ )  $\leftarrow$  ( $S_{ini}, F_{ini}$ )
3:   do
4:     ( $S_{min}, S_{max}, is\_valid$ )  $\leftarrow$  OPTIMIZEFREQUENCY( $S_i, F_i$ )
5:      $S_i \leftarrow S_{min} * 0.8 + S_{max} * 0.2$ 
6:      $F_i \leftarrow F_{new}$  with  $F_{new} < F_i$ 
7:   while  $is\_valid$ 

8: function OPTIMIZEFREQUENCY( $S_{start}, F$ )
9:   ( $S_{temp}, is\_valid$ )  $\leftarrow$  ( $S_{start}, true$ )
10:  do
11:     $S_{min} \leftarrow S_{temp}$ 
12:    ( $S_{temp}, is\_valid$ )  $\leftarrow$  run_cma( $S_{min}, F, min$ )
13:  while  $\|S_{temp} - S_{min}\| < \eta$  &&  $is\_valid$ 
14:  if  $is\_valid$  then
15:    do
16:       $S_{max} \leftarrow S_{temp}$ 
17:      ( $S_{temp}, is\_valid$ )  $\leftarrow$  run_cma( $S_{max}, F, max$ )
18:    while  $\|S_{temp} - S_{max}\| < \eta$  &&  $is\_valid$ 
19:  return ( $S_{min}, S_{max}, is\_valid$ )

```

---

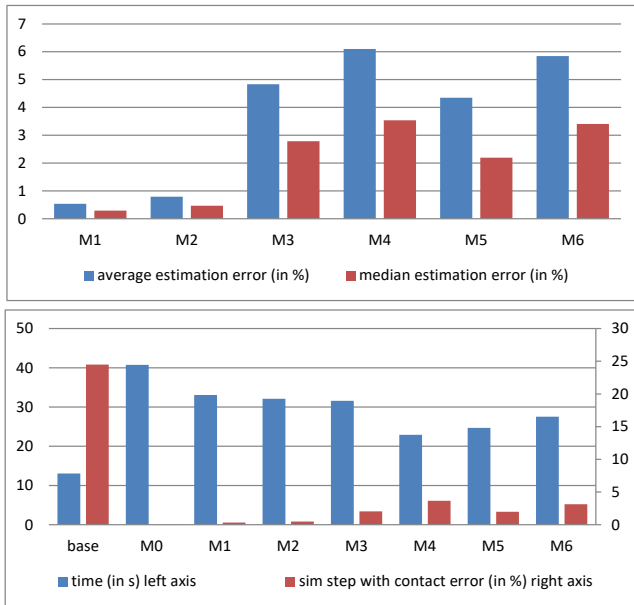
The desired coronal speed cycled between  $0m.s^{-1}$ ,  $0.2m.s^{-1}$  and  $-0.2m.s^{-1}$  every 5 character steps to introduce perturbation.

We first present the error in the evaluation of the future time steps by using various reduced models ( $M1$  to  $M6$ ). This error is the average difference in the GRF over 200 character steps between the reduced model and the full character model ( $M0$ ). We express it in percents of the sum of the GRF obtained with  $M0$ . The error for one time step is the sum of the difference in the normal component of the GRF on each corners of the foot and on the toes between the reduced model and the full character model.

The removed body parts in the six different reduced models are: arms ( $M1$ ), arms and head ( $M2$ ), everything above the pelvis ( $M3$ ), everything above the pelvis and the swing leg ( $M4$ ), arms, head and swing leg ( $M5$ ). We also tested a dynamic configuration ( $M6$ ):  $M3$  when both feet touch the ground and  $M4$  if only the stance foot touches the ground. As seen in Figure 1 (top), simplifying the arms and the head does not degrade the stabilizer.  $M6$  shows that the swing leg has also an important impact on the stabilizer. Finally, the comparison of  $M3$  and  $M5$  shows that simplifying the swing leg has a less negative impact than simplifying the torso.

Our second evaluation consists in comparing contacts improvement with required computation time. This evaluation was also done with the fully simulated character ( $M0$ ). As seen in Figure 1 (bottom), the number of simulation steps presenting a disequilibrium is greatly lowered by our stabilizer for all reduced models ( $M1$  to  $M6$ ). The comparison of  $M3$  and  $M5$  confirms the usefulness of removing the swing leg over the torso as  $M5$  has a smaller computation time for less steps showing a disequilibrium.

Two of the reduced models can be considered for use in the final system.  $M2$  results in a precise error correction without superfluous computations and  $M5$  offers a trade-off between computation time and error correction.



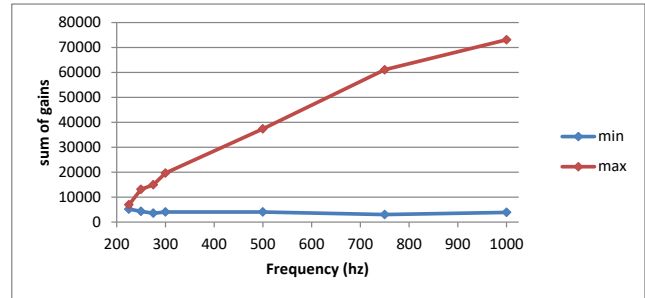
**Figure 1: Top: Estimation error for each reduced model (% of the original value); Bottom: red: percentile of the simulation steps presenting a disequilibrium of the GRFs, blue: time consumed for the evaluation. 'base' represents a controller where the ground contact stabilizer has been deactivated.**

## 5.2 Gains study

We initiated the offline gain optimization with the values from [17] for 1 kHz. We placed the target position for the hands just in front of the torso. We introduced perturbations by modifying the desired coronal speed.

The results seen in Figure 2 validate our hypothesis. We see that lower frequencies have more restrictive ranges for the acceptable gains. The optimization procedure was able to find acceptable gains for frequencies as low as 225 Hz while reproducing accurately the high frequency motion. However, at frequencies lower than 300 Hz, the controllers were not robust to large external perturbations anymore but could still be used to obtain a fast preview of the motion, i.e. for designing key poses.

Figure 3 (top) shows that the previously observed decrease of gains with the frequency is also observed for each individual joint with the exception of the hip. We must note that the root and torso\_head average values for 750 Hz are quite low compared to what we could have expected. By looking at the range of the values relative to the average (Figure 3 bottom) we can observe two phases. Starting at high simulation frequency and down to a certain frequency the range of possible values (in percents) stays constant. After that particular frequency, the range will either slowly or abruptly diminish. It is interesting to note that the end of the first phase is not observed at the same frequency for each joint. For most joints it is observed at frequencies of 300 Hz or less but seems to happen above 1000 Hz for the knee and ankle. The hip seems to have a behaviour similar to those two with the values for 275



**Figure 2: Representation of the sum of the gains  $\sum K_p + K_d * 10$  corresponding to the minimum (blue) and the maximum (red) for each tested frequency.**

Hz and 1000 Hz seemingly being outliers. The hip seems to be a special joint. It has a constant and low average value across all simulation frequencies and has a small range (smaller than 20% of the average value) for every frequency. The small range may be observed because the hip has two roles depending on the stance and it affects an important part of the character in each stance. The presence of small average values is also observable on the ankle and the toes joint. Since the toes joint only affects the foot and the toes, which are low mass body parts, in either stance, low gains values are expected. The low values on the hip and the ankle are most likely due to the fact that those two joints are the ones the most affected by the balance and velocity control systems.

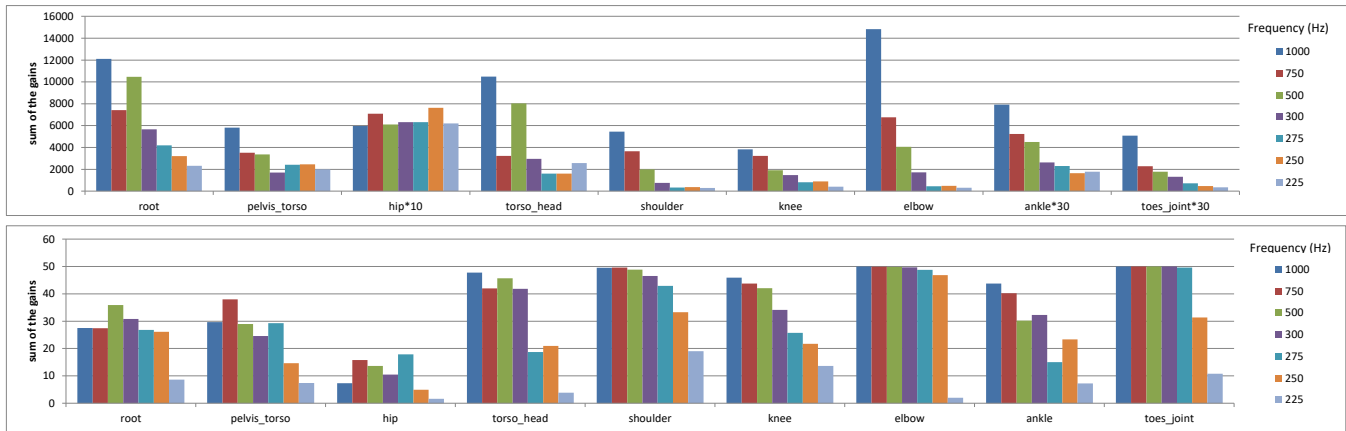
## 5.3 Discussion

Our ground contact stabilizer is not yet capable of correcting for a large disequilibrium of the GRF. In particular, correcting contacts when only one corner of the foot cuboid touches the ground is currently impossible. This situation might appear when applying large external forces at low frequencies. To prevent wasting time on such case, we limit our online CMA to five iterations. When testing lateral pushes for lower simulation frequencies, up to 6% of the simulations step might trigger that limit.

The reported performance for the ground contact stabilizer have been obtained with a single thread implementation. As the estimations of the samples inside one step of our online CMA are completely independent from each other, we could evaluate them simultaneously in separate threads. Our controller usually needs one or two iterations of the CMA, meaning we could save up to 75% of the time currently used by the stabilizer.

The performance depends on the starting point used for the CMA. Even though the rules we used to determine starting points are working for a wide range of frequencies, few more iterations of the CMA are sometimes needed causing an increase of computation time.

The controllers obtained with frequencies lower than 300 Hz are not robust to large external perturbations even though still more robust than the ones reported in previous works that use a similar control strategy.



**Figure 3: top: Evolution of the average sum of the gains  $\sum K_p + K_d * 10$  for each frequency and joint, to allow every values to be read on the same histogram the values for the hip have been multiplied by 10, and the ankles and toes by 30; bottom: Evolution of the range of the sum of the gains  $\sum K_p + K_d * 10$  for each frequency and joint (% of the average value).**

## 6 CONCLUSION

In conclusion, we have designed and evaluated a controller capable of simulating the locomotion of a virtual character with control frequencies down to 300 Hz which is lower than the ones presented in previous works for this type of controller. Our controller allows frequencies down to 225 Hz if no external perturbation is applied on the character. We have demonstrated the ability to realize up to 115° turns by managing the change of heading over a small period of time. We also proposed a system reducing the perturbations caused by the loose contacts between the stance foot and the ground at low frequencies. We did not rely on a particular physics engine, the presented components are all part of the controller.

In the future, we would like to investigate a method to help the ground contact stabilizer to produce a stable motion even when extreme contact conditions are detected. Having a multi-threaded implementation would free a lot of computation time. Also we would like to see if our observations on the effect of frequency on gains would allow to find gains at low frequencies for other types of motion such as running. We could implement a system using biomechanical data to determine the poses to track in order to achieve more natural looking sharp turns. Finally, changing the PD-controller by Stable PD-controllers (SPD) proposed by [14] could allow our system to reach even lower frequencies as their controllers make high gains more stables.

## REFERENCES

- [1] Samuel Carensac, Nicolas Pronost, and Saïda Bouakaz. 2015. Real-time gait control for partially immersed bipeds. In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*. ACM, 177–182. <http://dl.acm.org/citation.cfm?id=2822016>
- [2] Stelian Coros, Philippe Beaudoin, and Michiel Van de Panne. 2010. Generalized biped walking control. *ACM Transactions on Graphics (TOG)* 29, 4 (2010), 130. <http://dl.acm.org/citation.cfm?id=1781156>
- [3] T. Geijtenbeek and N. Pronost. 2012. Interactive Character Animation Using Simulated Physics: A State-of-the-Art Review. *Computer Graphics Forum* 31, 8 (Dec. 2012), 2492–2515. <https://doi.org/10.1111/j.1467-8659.2012.03189.x>
- [4] Stevie Giovanni and KangKang Yin. 2011. LocoTest: deploying and evaluating physics-based locomotion on multiple simulation platforms. *Motion in Games* (2011), 227–241. <http://link.springer.com/content/pdf/10.1007/978-3-642-25090-3.pdf#page=238>
- [5] David Andrew Greer. 2016. *Physics-based character locomotion control with large simulation time steps*. Ph.D. Dissertation. Bournemouth University. <http://eprints.bournemouth.ac.uk/25038/>
- [6] Daseong Han, Haegwang Eom, Junyong Noh, and others. 2016. Data-guided Model Predictive Control Based on Smoothed Contact Dynamics. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 533–543. <http://onlinelibrary.wiley.com/doi/10.1111/cgf.12853/full>
- [7] Nikolaus Hansen. [n. d.]. *The CMA Evolution Strategy: A Comparing Review*.
- [8] Sumit Jain and C. Karen Liu. 2011. Controlling physics-based characters using soft contacts. In *ACM Transactions on Graphics (TOG)*, Vol. 30. ACM, 163. <http://dl.acm.org/citation.cfm?id=2024197>
- [9] Libin Liu, Michiel Van De Panne, and Kangkang Yin. 2016. Guided Learning of Control Graphs for Physics-Based Characters. *ACM Transactions on Graphics* 35, 3 (May 2016), 1–14. <https://doi.org/10.1145/2893476>
- [10] Libin Liu, KangKang Yin, Michiel van de Panne, Tianjia Shao, and Weiwei Xu. 2010. Sampling-based contact-rich motion control. In *ACM Transactions on Graphics (TOG)*, Vol. 29. ACM, 128. <http://dl.acm.org/citation.cfm?id=1778865>
- [11] Libin Liu, KangKang Yin, Bin Wang, and Baining Guo. 2013. Simulation and control of skeleton-driven soft body characters. *ACM Transactions on Graphics (TOG)* 32, 6 (2013), 215.
- [12] Uldarico Muico, Yongjoon Lee, Jovan PopoviĀG, and Zoran PopoviĀG. 2009. Contact-aware nonlinear control of dynamic characters. In *ACM Transactions on Graphics (TOG)*, Vol. 28. ACM, 81. <http://dl.acm.org/citation.cfm?id=1531387>
- [13] Uldarico Muico, Jovan PopoviĀG, and Zoran PopoviĀG. 2011. Composite control of physically simulated characters. *ACM Transactions on Graphics* 30, 3 (May 2011), 1–11. <https://doi.org/10.1145/1966394.1966395>
- [14] Jie Tan, Karen Liu, and Greg Turk. 2011. Stable proportional-derivative controllers. *IEEE Computer Graphics and Applications* 31, 4 (2011), 34–44.
- [15] Yuval Tassa, Tom Erez, and Emanuel Todorov. 2012. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 4906–4913.
- [16] Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. MuJoCo: A physics engine for model-based control. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 5026–5033. <http://ieeexplore.ieee.org/abstract/document/6386109/>
- [17] KangKang Yin, Kevin Loken, and Michiel Van de Panne. 2007. Simbicon: Simple biped locomotion control. In *ACM Transactions on Graphics (TOG)*, Vol. 26. ACM, 105. <http://dl.acm.org/citation.cfm?id=1276509>