In Proceedings of 8th Workshop on Humanoid Soccer Robots
13th IEEE-RAS International Conference on Humanoid Robots (Humanoids), Atlanta, GA, 2013.

Simultaneous evolution of leg morphology and walking skills to build the best humanoid walker

Nicolas Jouandeau
LIASD
Paris8 University
Email: n@ai.univ-paris8.fr

Vincent Hugel
LISV
Versailles University
Email: hugel@lisv.uvsq.fr

***Abstract*—This paper presents an optimization process designed for simultaneous tuning of humanoid's physical characteristics and walking skills' parameters. Starting from the NAO's model used in the 3D Simulation Soccer League, this process focuses on 5 specific parameters to improve individual walking performances. Two policies allow to generate best-first agent or best-average agent. Further improvements may be possible by addressing more parameters or by linking improvements to existing high-level analysis. With these first physical modifications, we achieved to walk 1.5 times faster than the original NAO. In addition results produce more realistic, safer and more precise walk.**

I. Introduction

Even for humanoids, athletic performances can be evaluated by comparing ranking with others or simply measuring distances, times or scores as for humans. During their training process, performers need an evaluation of their results to set the next tests that will be determining. In this paper, we focus on individual training by improving physical properties and technical skills of a humanoid robot. In other words the physical model extracted from humanoid's characteristics, and the moves' primitives defined by humanoid's skills are being evolved simultaneously to build the best humanoid performer. Recent work [1] was developed to automatically generate parameters for the geometric modelling of kinematic chains, which can be used to address the control issue of heterogeneous teams of humanoids. In the same time, the RoboCup 3D Simulation Soccer League (3DSSL) made three different humanoid players available to build heterogeneous teams. In addition to the heterogeneity of players, we show that we can optimize humanoids' morphology and walking skills altogether. Section II is dedicated to related work. Section III presents the optimization process proposed. Section IV details experimental results. Section V deals with the conclusion.

II. Related work

During the last RoboCup 2013 competition, the 3D Simulation Soccer League allowed competitive teams to experiment heterogeneous humanoid's cooperation. The competition rules admit the use of 3 different humanoid's models with various leg's physical models, elbow's physical models and foot's joints maximum speeds parameters. In this competition, humanoid robots play soccer games. Each team has 11 players. Individual performances and team play are essential. Therefore, it is interesting to extend classical model performances with new physical models, that could lead to specializations of players.

More generally, the problem of model optimization is always posed for each new humanoid. The number of joints is fixed according to costs and expected mobility. Then the manufacturing process depends on joints shape definitions, motors sizing, and geometry. The definition of geometric parameters can be actually enhanced thanks to physical tuning to optimize humanoid's performances (e.g. in the last NAO version, legs are slightly longer). Along this process, excessive complexity can arise from individual capabilities optimization. The Denavit Hartenberg [2] convention is well appreciated for its uniform formalism in kinematic chains modelling. This convention modified by Khalil Kleinfinger [3] was used to automatically compute NAO's geometric parameters [1], which makes automatic tuning of humanoid physical parameters possible.

To face costs of humanoid platforms, heterogeneous cooperation of humanoids systems has been used in cleaning or assisting missions instead of abstract humanoid's models [4]. Since differences between heterogeneous agents can lead to specific optimizations, it is important to define a level of abstraction. We believe that a learning process must automatically rule the optimization of lowest levels. By this way, the solution provided could deal with cooperation and heterogeneity in very large humanoids teams without decreasing performances. Heterogenous humanoids' systems should also be able to make richer interactions in large multi-agent systems than homogeneous systems.

Ordered from high level to low level, here is a list of previous works that aimed at improving humanoid soccer teams' behaviour:

- Team coordination mechanisms optimization over coaching architectures [5] where each team member's position is optimized with flexible formations based on situations' identification.
- Role allocation [6] that optimizes utility functions that assign agents to roles in a classical matrix form of Optimal Assignment Problem.
- Procedural role/positioning coupling [7] that optimizes covering surface of the team while reducing the team size.
- Interdependent skills optimization [8] that optimizes walking and kicking skills taking into account speed,

placement and skills cooperation inside game sequences.

- Learning walking skills from real behaviour's [9] that shows relations between simulation and real world optimizations.
- Walk engine parameters optimization [10] that optimizes walking parameters for best speed and stability.
- Gait optimization [11] that optimizes sets of parameters according to walking directions by using Particle Swarm Optimization.

According to recent work [1] on automatic generation of humanoid's geometric modelling parameters, low level optimization is possible to enhance individual morphology and walking skills according to desired needs.

III. Humanoid's optimization process

In this section, we present the evolving process used to define what can benefit humanoids in terms of displacement capabilities. We actually describe the optimization process that is performed while physical characteristics and parameters' skills are being evolved.

A. Optimizing parameters

To settle benefits in humanoid growth, we choose the black box optimizer CLOP [12] (that stands for Confident Local OPtimization). Then the problem of tuning parameters becomes a decision process to establish what are better, equivalent and worse results. CLOP is an iterative process that mainly needs:

- A list of input parameters.
- Lower and upper bounds for each parameter.
- A decision function that states outcome of a single CLOP iteration.

The learning approach developed in [10] that uses the nonsmooth optimization stochastic solver CMA-ES where presumably similar sensitivity input variables are expected. Unlike this approach, we use the CLOP technique that performs a noisy smooth optimization, which is more suited to heterogeneous variables with faster convergence than CMA-ES.

Input parameters and their bounds are defined in the set $\mathcal{L}$. At each iteration of the black box optimization process, a new set of parameters and values are chosen from $\mathcal{L}$ according to previous iterations in order to maximize next iterations success rates. This is realized by balancing uniform space sampling and previous iterations' results. The evolving process consists of observing the function that measures probability performances of samples p from $\mathcal{L}$ and adapting samples to expected results. During the evolving process, samples are populating the history set $\mathcal{H}$. At each step, each previous sample is reconsidered according to other samples. This evaluation is done with a regression over all samples, by computing average strength, expected results and weighting samples.

To consolidate the evaluation process, we define a single CLOP iteration as a set of similar trials that are performed multiple times. The objective is to settle moves that are based on reliable assessment. Moves have to be robust enough even if they are faster. Thus a CLOP iteration is defined by multiple trials with the same set p. Each CLOP iteration produces averaged values for these trials. At the end of each CLOP iteration, the decision function `pickOut` (see §*B*) states if the result is better, equivalent or worse than the best known result. Then parameters converge to best evolving values.

B. Simply evolving

The evolving optimization process is presented in Alg. 1. The balance between sampling new values and weighting previous results is done by the history set $\mathcal{H}$ that is fed by `pickOut` and used as input by `newParams`. The `pickOut` function returns the three possible values $ACCEPT$, $EQUIVALENT$ and $REJECT$ (that correspond to better, equivalent and worse results). At each iteration, a new set of parameters p are chosen. ν' stands for best acceptable results. At the beginning, ν' is an empty set. Each CLOP iteration implies that a new tuple (h,p) is inserted in $\mathcal{H}$. If n is too small, ν' could remain empty, thus meaning that no solution had been found over n iterations. As presented in Alg. 1 line `1`, the process is started from scratch. The process could also be started from an expert knowledge base, by removing this line and setting ν' to specific values. The process could also be started with a previous knowledge base sampling, setting a previous $\mathcal{H}$ experiment value to accumulate skills or to vary parameters' space sampling.

Algorithm 1 `evolving` (n, $\mathcal{L}$, `pickOut`)

1: $(\nu', \mathcal{H}) \leftarrow (\emptyset, \emptyset, \emptyset)$
2: **for** $i = 0$ to n **do**
3: $\quad p \leftarrow$ `newParams` $(\mathcal{H}, \mathcal{L})$
4: $\quad (s, \nu) \leftarrow$ `multipleTrials` (p)
5: $\quad (\nu', h) \leftarrow$ `pickOut` (s, ν, ν')
6: $\quad$ `insert` $((p\ , h), \mathcal{H})$
7: **end for**
8: **return** `paramsFrom` (ν')

The `pickOut` function takes 3 parameters s, ν, ν' that are:

- The success rate s of the last experiment that is equal to the number of trials without fall, divided by the total of trials.
- The set ν that contains last experiment results.
- The set ν' that contains the best known experiment results.

IV. Experiments

Tests are run on a simple computer. The experiment software system is composed of 5 parts, i.e. *rcssserver3d* [14], [15], our client agent *rcssagent3d-l3m* [13], a coach (that is responsible for starting trial), the CLOP framework [12] and utilities that link evolving optimization process to CLOP.

We present two experiments, in the first one we perform a simple physical evolution, and in the second one, a more sophisticated evolution. The first experiment aims at checking the proper sizing of the NAO's model. The second experiment changes physical parameters and technical skills simultaneously to fit morphology and walking parameters to maximize speed while minimizing lateral drift.

A. Simple physical evolution

Using the NAO model provided by RoboCup 2013 3DSSL, the simple growth consists of varying the relative distance between hip and thigh center of mass from -0.01 to $-0.10[m]$. The objective is to walk forward faster, straight and avoid falling. Each CLOP iteration performs 10 trials. Each trial requires the simulated robot to go forward until $10[sec]$.

The internal decision function process is detailed in Alg. 2. Inside each set ν, the subset m defines average values and the subset e defines standard deviations values. Condition line `1` checks if the robot does not fall too much. Condition line `4` checks if NAO is walking straight enough. Then, if condition line `7` is met, current parameters are the first best solution. Once this first good enough result is found, m' and e' are no longer empty. Then the three static parameters α, β and γ are taken into account :

- The nearness factor α: if the new result is not close enough to the best known result, parameters are considered to lead to a worst instead of a best known result.
- The equivalence factor β: the test is similar to the nearness factor test with a different threshold. As the result is now $EQUIVALENT$ with this factor and is $REJECT$ with α, as it stands for comparing averages and standard deviations results of two experiments, it is logical to ensure that $\beta < \alpha$.
- The width factor γ: if the new result is more stable, then it is better.

After the use of these three factors (lines `10`, `13` and `16`), the last test checks whether the result can be considered as equivalent to the previous best known result if the axis-align-translation value m_x is less than this previous result. As results with smaller standard deviations have ever been accepted in line `16`, line `19` discards results that are close by average and have larger standard deviations. As the `pickOut` function is iteratively used, the values $SUCCESS\_RATE$, XY_RATIO, α, β and γ contribute to the convergence speed of the evolution.

B. Physical and technical evolution

Here we simulate a nature-inspired growth by simultaneously evolving structural parameters of the legs (Tab. I) and locomotion parameters (Tab. II). The walking primitives used have a fixed step size, except for the first two steps that are two times shorter.

TABLE I. LEG MORPHOLOGY PARAMETERS INTERVALS

$ThighRelHip2\_Z$	[-0.08;-0.03]
$ratio\_flexion$	[0.60; 0.95]

$ThighRelHip2\_Z$ and $ratio\_flexion$ parameters presented in Tab. I are related to the leg morphology, which are the semi-length of the femur and the leg flexion ratio. By changing the semi-length of the femur, we vary the cural index of the leg, which is the ratio of the tibia length with the femur length. This index has a great importance in human morphology since it is used to compare the different bipeds that colonized the Earth.

The flexion ratio is defined as the ratio of the hip height from the ground over the total length of the leg when stretched. By changing the flexion ratio, the robot can walk with knees more or less flexed.

The three other parameters are used to adjust the walking skills of the robot, in order to have a well balanced and a quick gait. These parameters, presented from top to bottom in Tab. II, are the horizontal distance between the middle of the ankles and the torso center, the maximal height of leg lift-off, and the maximal step length.

The horizontal distance between the middle of the ankles and the torso center allows to balance the weight of the torso with respect to the flexed legs. The COM is considered to be fixed with respect to the torso, and its coordinates inside the torso coordinate frame are calculated automatically in the standing position as a function of the morphological parameters. This is an usual approximation in the case of the LIP-3D model.

Algorithm 2 `pickOut` (s, m, e, m', e')

1: **if** $s < SUCCESS\_RATE$ **then**
2:     **return** $REJECT$;
3: **end if**
4: **if** $m_y/m_x > XY\_RATIO$ **then**
5:     **return** $REJECT$;
6: **end if**
7: **if** $m' == UNDEFINED$ **then**
8:     **return** $ACCEPT$;
9: **end if**
10: **if** $m_x < m'_x - \alpha e'_x$ **then**
11:     **return** $REJECT$;
12: **end if**
13: **if** $m_x < m'_x - \beta e'_x$ **then**
14:     **return** $EQUIVALENT$;
15: **end if**
16: **if** $e_x < \gamma e'_x$ **then**
17:     **return** $ACCEPT$;
18: **end if**
19: **if** $m_x < m'_x$ **then**
20:     **return** $EQUIVALENT$;
21: **end if**
22: **return** $ACCEPT$;

TABLE II. WALKING SKILLS PARAMETERS INTERVALS

$long\_offset\_MidAnkles\_2\_Torso\_Init$	[0.001;0.030]
$height\_lift$	[0.025;0.070]
$xlength\_step\_max$	[0.020;0.150]

V. RESULTS

Table III shows decision parameters used for all experiments. The first two parameters come from our experience in 3DSSL competitions, where standing posture rate is fixed to 75% (*i.e.* tolerating 25% of falls) and 25% of lateral drift are admitted. Parameters α and β are standard decisive values for normal distribution. At last, the parameter γ is a ratio to quantify more stable results.

TABLE III. pickOut DECISION PARAMETERS

$SUCCESS\_RATE$	0.75
XY_RATIO	0.25
α	3.0
β	1.0
γ	0.7

After analysing the results from the two experiments, we carry out a last experiment to compare optimization process parameters with our previous parameters.

A. Simple physical evolution results

Figure 1 shows $ThighRelHip2\_Z$ evolution's results over 500 iterations. $ThighRelHip2\_Z$ results are represented on the vertical y-axis and vary between -0.1 and $-0.01[m]$. Iterations are represented horizontally. It contains 36 $REJECT$ (represented in black), 177 $EQUIVALENT$ (represented in gray) and 287 $ACCEPT$ (represented in white). It shows the results' evolution according to the chosen $ThighRelHip2\_Z$. As the first 50 iterations contain 28 $REJECT$, 18 $EQUIVALENT$ and 3 $ACCEPT$, and the next 50 iterations contain 6 $REJECT$, 18 $EQUIVALENT$ and 26 $ACCEPT$, it shows that $ThighRelHip2\_Z$ values are rapidly converging to the $ACCEPT$ interval.

The first 100 values of $ThighRelHip2\_Z$ are detailed in the Fig. 2. It shows more clearly that values are rapidly converging to the interval $[-0.05;-0.025]$.

The mean of best values after 500 iterations is $-0.038[m]$. It confirms that the actual NAO's model value of $-0.04[m]$ is correctly sized.

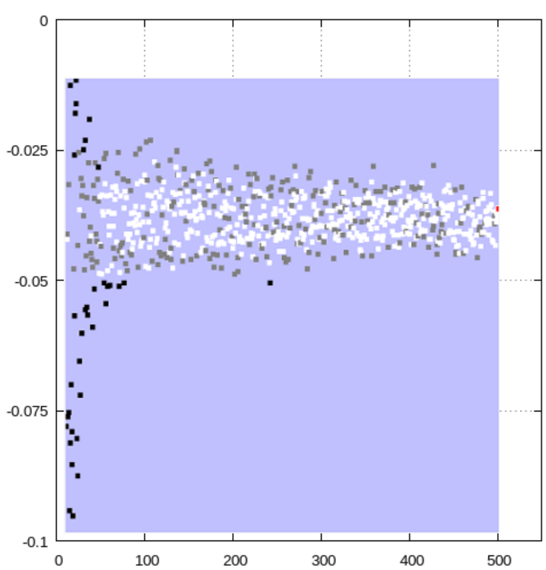


Fig. 1. $ThighRelHip2\_Z$ results for 500 iterations where better results are white, equivalent results are gray and worst results are black.

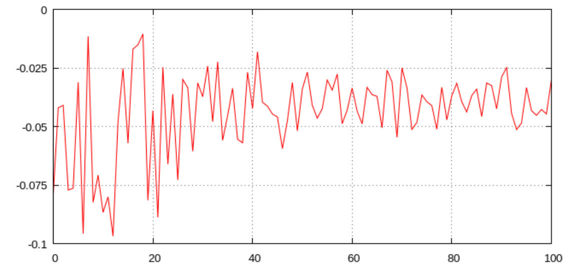


Fig. 2. $ThighRelHip2\_Z$ results for the first 100 iterations.

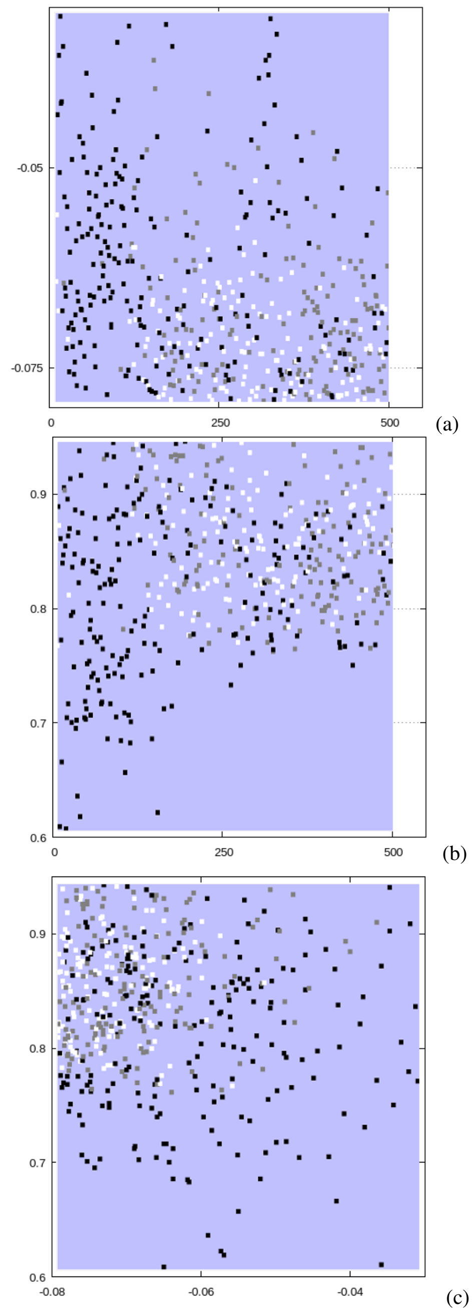
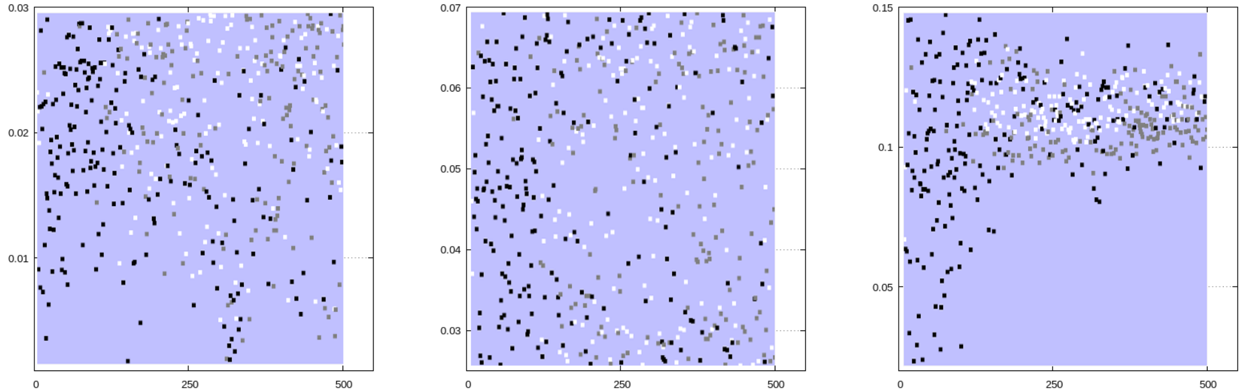
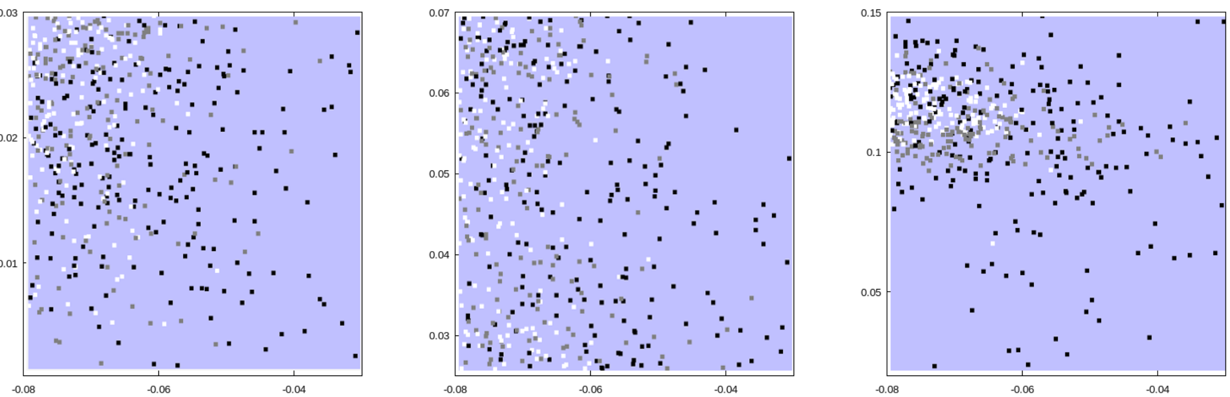


Fig. 3. (a) Results of $ThighRelHip2\_Z$ from 0 to 500 iterations. (b) Results of $ratio\_flexion$ from 0 to 500 iterations. (c) $ratio\_flexion$ (y-axis) over $ThighRelHip2\_Z$ (x-axis) results.



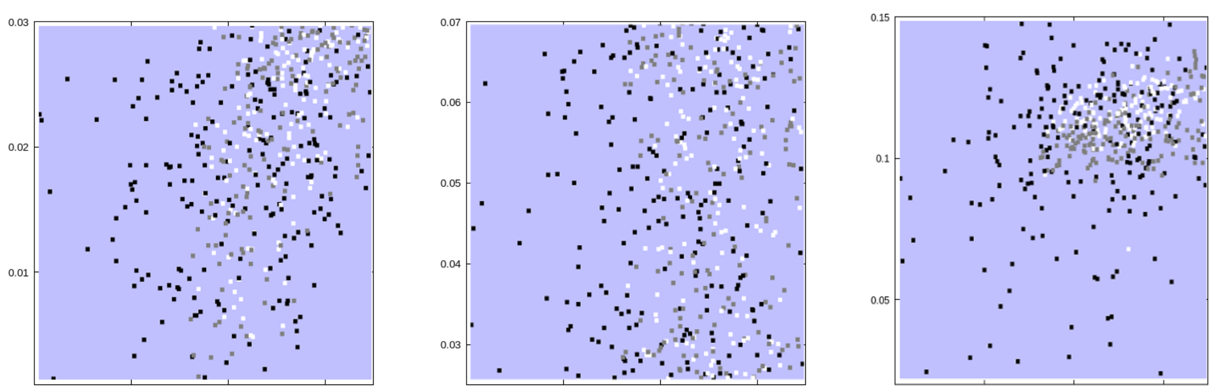
(a) (b) (c)

Fig. 4. $long\_offset\_MidAnkles\_2\_Torso\_Init$ (a), $height\_lift$ (b) and $xlength\_step\_max$ (c) results for 500 iterations.



(a) (b) (c)

Fig. 5. $ThighRelHip2\_Z$ (x-axis) over $long\_offset\_MidAnkles\_2\_Torso\_Init$ (y-axis on (a)), over $height\_lift$ (y-axis on (b)) and over $xlength\_step\_max$ (y-axis on (c)) results.



(a) (b) (c)

Fig. 6. $ratio\_flexion$ (x-axis) over $long\_offset\_MidAnkles\_2\_Torso\_Init$ (y-axis on (a)), over $height\_lift$ (y-axis on (b)) and over $xlength\_step\_max$ (y-axis on (c)) results.

B. Physical and technical evolution results

According to bound intervals mentioned in Tab. I and II, Fig. 3a and 3b show physical evolution for 500 iterations, while walking skills parameters evolution are shown in Fig. 4a, 4b and 4c.

During the first 100 iterations, $REJECT$ results dominate. Following these iterations, $ThighRelHip2\_Z$, $ratio\_flexion$ and $xlength\_step\_max$ are clearly converging to $\mathcal{L}$ subspaces. The two others parameters $long\_offset\_MidAnkles\_2\_Torso\_Init$ and $height\_lift$ stay uniformly distributed in $\mathcal{L}$. At the end of the optimization process, $ACCEPT$ and $EQUIVALENT$ results dominate for all parameters.

Figures 3c, 5 and 6 show inter parameters relations :

- between the two physical parameters $ThighRelHip2\_Z$ and $ratio\_flexion$ on Fig. 3c.
- between walking skills parameters and $ThighRelHip2\_Z$ on Fig. 5.
- between walking skills parameters and $ratio\_flexion$ on Fig. 6.

It confirms that there is no simple one-to-one but multiple relationship between these parameters. It also shows that $xlength\_step\_max$ is linearly dependent on the two physical parameters.

C. Comparing evolution results with previous results

Table IV shows results for 100 trials with parameters' values that result from optimization process. The Expert column contains physical parameter values, among them ($ThighRelHip2\_Z = -0.040[m]$) used in the last 3DSSL RoboCup competition [13], and the other four parameters that we set according to our knowledge expertise. The *Optim.1* column stands for best last trial of the `pickOut` decision function used in our humanoid's optimization process. *Optim.1* builds a best-first agent. The *Optim.2* column stands for weighed values from the best last trials. *Optim.2* builds a best-average agent. *Optim.1* and *Optim.2* have better success rates than Expert. Compared to Expert, the average forward distance gain is 1.5 for *Optim.1* and 1.4 for *Optim.2*, and lateral deviation is nearly the same for *Optim.2*.

TABLE IV. RESULTS FROM PHYSICAL AND WALKING SKILLS OPTIMIZED SETS FOR 100 TRIALS.

	Expert	*Optim.1*	*Optim.2*
$ThighRelHip2\_Z$	-0.040	-0.079	-0.073
$ratio\_flexion$	0.730	0.902	0.864
$long\_offset\_MidAnkles\_2\_Torso\_Init$	0.011	0.020	0.024
$height\_lift$	0.042	0.063	0.049
$xlength\_step\_max$	0.080	0.125	0.115
s	0.73	0.95	0.98
m_x	4.744	7.121	6.737
m_y	0.510	0.721	0.591
e_x	0.088	0.229	0.089

These results show that humanoids can benefit from longer legs (*i.e.* longer $ThighRelHip2\_Z$) by using longer steps (*i.e.* longer $xlength\_step\_max$). Evolutions *Optim.1* and *Optim.2* use longer thighs with higher $ratio\_flexion$. These two walking profiles produce more realistic walk. Actually legs are more stretched than our previous walk, which makes *Optim.1* and *Optim.2* walking gaits closer to human walk. The main advantages resulting from the physical transformations associated with walking parameters are increased walking speed (with larger m_x values), safer walking (larger s values, which means a reduced rate of falls) and more precise walking (reduced m_y/m_x ratio).

VI. CONCLUSION

We have presented an optimization process designed for humanoid's physical characteristics and walking skills. Two policies produce best-first agent and best-average agent, that improve the previous results based on knowledge expertise. The optimization process is simple and mainly guided by a decision function that distinguishes between better, equivalent and worse results. After a few iterations, the process converge to better values. While developing new models according to specific skills, we believe that more rich interactions and team play can be developed to settle stronger humanoid teams.

REFERENCES

[1] V. Hugel and N. Jouandeau, *Automatic generation of humanoid s geometric model parameters*, In 17th annual RoboCup International Symposium 2013 (RCUP-2013).

[2] J. Denavit and R.S. Hartenberg, *A kinematic notation for lower-pair mechanisms based on matrices*, In Trans ASME J. Appl. Mech. 23 pp. 215-221, 1955.

[3] W. Khalil and J.F. Kleinfinger, *A new geometric notation for open and closed-loop robots*, In IEEE Int. Conf. on Robotics and Automation, pp. 1174-1180, 1986, (ICRA'86).

[4] L. Heonyoung, K. Yeonsik, L. Joongjae, K. Jongwon, Y. Bum-Jae, *Multiple Humanoid Cooperative Control System for Heterogeneous Humanoid Team*, In IEEE Int. Symp. on Robot and Human Interactive Communication, Munich, Germany, 2008.

[5] N. Lau and L. Paulo Reis, *FC Portugal-high-level coordination methodologies in soccer robotics*, In Robotic Soccer Journal, pp. 167-192, 2007.

[6] B.P. Gerkey and M.J. Mataric, *On role allocation in RoboCup*, In RoboCup 2003: Robot Soccer World Cup VII, LNCS Vol. 3020, pp. 43-53, 2003.

[7] N. Lau, L. Seabra Lopes, G. Corrente and N. Filipe, *Multi-robot team coordination through roles, positionings and coordinated procedures*, In IEEE Int. Conf. on Intelligent Robots and Systems, 2009, (IROS 2009).

[8] D. Urieli, P. MacAlpine, S. Kalyanakrishnan, Y. Bentor and P. Stone, *On Optimizing Interdependent Skills: A Case Study in Simulated 3D Humanoid Robot Soccer*, In Int. Conf. Autonomous Agents and Multiagent Systems, pp. 769-776, 2011, (IFAAMAS-2011).

[9] A. Farchy, S. Barrett, P. MacAlpine and P. Stone, *Humanoid Robots Learning to Walk Faster: From the Real World to Simulation and Back*, In Int. Conf. on Autonomous Agents and Multiagent Systems, 2013, (AAMAS 2013).

[10] P. MacAlpine and P. Stone, *Using Dynamic Rewards to Learn a Fully Holonomic Bipedal Walk*, In Adaptive Learning Agents Workshop, 2012, (ALA 2012).

[11] C. Niehaus, T. Röfer and T. Laue, *Gait Optimization on a Humanoid Robot using Particle Swarm Optimization*, In Second Workshop on Humanoid Soccer Robots, IEEE-RAS 7th International Conference on Humanoid Robots, 2007.

[12] R. Coulom, *CLOP: Confident Local Optimization for Noisy Black-Box Parameter Tuning*, In Advanced Computer Games (ACG), volume 7168 of Lecture Notes in Computer Science, page 146-157. Springer, 2011.

[13] N. Jouandeau, V. Hugel and T. Da Costa, *3DSSL Team Description Paper*, In Robocup 3D Soccer Simulation League 2013, (RCUP-2013).

[14] O. Obst and M. Rollmann, *Spark - A Generic Simulator for Physical Multi-Agent Simulations*, In Multiagent System Technologies, Lecture Notes in Computer Science, vol. 3187, pp. 243-257, 2004.

[15] *SimSpark, a generic physical multiagent simulator system for agents in three-dimensional environments*, http://simspark.sourceforge.net/.