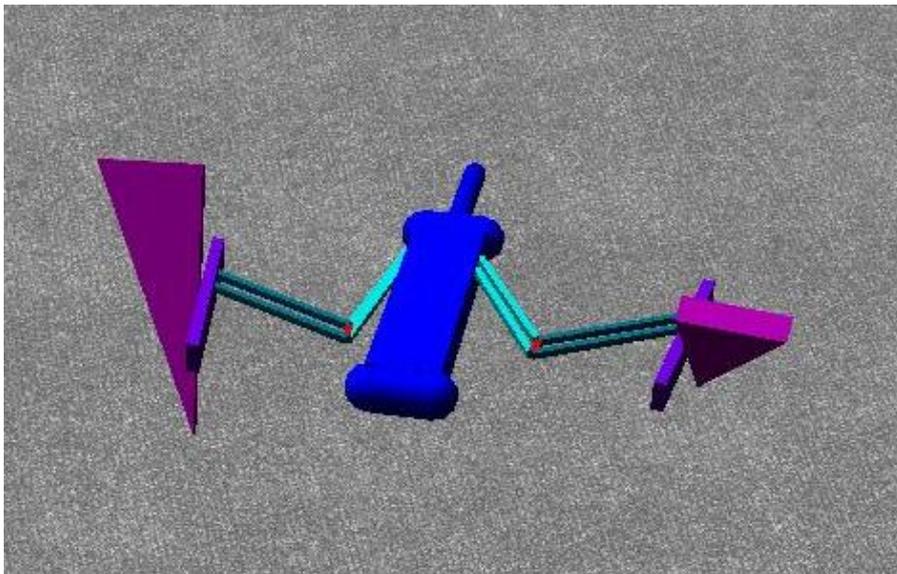




ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Semester project
LIS - Laboratory of Intelligent Systems

**Evolution of a Ground Controller for a Flying and Walking Robot
with Adaptive Morphology**



Noémie Jaquier

Professor: Dario Floreano
Assistants: Ludovic Daler, Josh Auerbach

Autumn 2014

Official project description

EIDGENÖSSISCHE TECHNISCHE HOCHSCHULE LAUSANNE
POLITECNICO FEDERALE DI LOSANNA
SWISS FEDERAL INSTITUTE OF TECHNOLOGY LAUSANNE

FACULTE SCIENCES ET TECHNIQUES DE L'INGENIEUR

LABORATORY OF INTELLIGENT SYSTEMS (LIS)
CH - 1015 LAUSANNE



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

SEMESTER PROJECT

Title: An Evolved Controller for a Flying and Walking Robot with Adaptive Morphology

Student(s): Noémie Laure Gwendoline Jaquier (MT)

Professor: Dario Floreano

Assistant 1: Ludovic Daler

Assistant 2: Josh Auerbach

Project description:

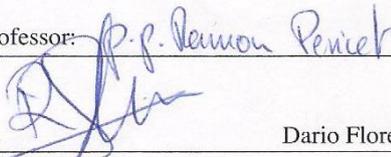
At the LIS we are developing a novel flying platform which has the ability to both move easily through the air and on the ground. This new platform will be able to hover, fly, walk on the ground, and upright itself thanks to its wings. These abilities bring this new flying robot closer to the capabilities of birds that are much more adaptive to their environment than current flying robots.

The goal of this project is to enable a robot, which can fly forward like a plane and also walk on the ground using its wings, with autonomous capabilities. The current version of the robot also has deployable wings which allow him to change its morphology in order to be more efficient while moving on the ground. Furthermore, this deployable mechanism can be used during walking to increase the steps of the robot. A first controller has been implemented to control the robot on the ground. The goal of this project is to improve this controller in order to control as well this deployable mechanism during walking. To do so, the robot will be simulated in an existing physics based simulator. Then, the synchronization of the motors used for the deployable mechanism and for the walking will be evolved in the simulator in order to find an optimal gait. The final goal of this project will be to show that the robot can evolve different gaits depending on the terrain. Finally, depending on the advancement of the project, these different gaits will be tested on the real robot.

Remarks:

You should present a research plan (Gantt chart) to your first assistant before the end of the second week of the project. An intermediate presentation of your project, containing 8 minutes of presentation and 7 minutes of discussion, will be held on October 29, 2014. The goal of this presentation is to briefly summarize the work done so far and discuss a precise plan for the remaining of the project. Your final report should start by the original project description (this page) followed by a one page summary of your work. This summary (single sided A4), should contain the date, laboratory name, project title and type (semester project or master project) followed by the description of the project and 1 or 2 representative figures. In the report, importance will be given to the description of the experiments and to the obtained results. A preliminary version of your report should be given to your first assistant at the latest 10 days before the final hand-in deadline. 2 copies of your final version, signed and dated, should be brought to your first assistant before noon January 9, 2015. A 20 minute project defense, including 5 minutes for discussion, will take place between January 12 and January 23, 2015. You will be graded based on your results, report, final defense and working style. All documents, including the report (source and pdf), summary page and presentations along with the source of your programs should be handed in on a CD on the day of the final defense at the latest.

Responsible professor:


Dario Floreano

Responsible assistant:

Signature:



Ludovic Daler

Lausanne, 16 September 2014

SEMESTER PROJECT – Autumn 2014

Title :	Evolution of a Ground Controller for a Flying and Walking Robot with Adaptive Morphology		
Candidate :	Noémie Jaquier	Section :	Microtechnique
Professor :	Dario Floreano	Assistant 2 :	Josh Auerbach
Assistant 1 :	Ludovic Daler		

Project Summary

The goal of this project is to improve the walking controller of a multi-modal bio-inspired walking and flying robot, the DALER.

The DALER rotates the extremity of its wings, called wingerons to push itself forward to walk on the ground. Two motorised joints, called shoulder and elbow, and one free joint on each wing allow them to be deployable.

This report treats the optimisation of the walk using also the deployable mechanism of the wings. The robot is modelled on the submicroscopic simulator Webots, as showed by Figure 1. Moreover, a genetic algorithm is used to evolve the parameters of the motors to find an optimal gait.

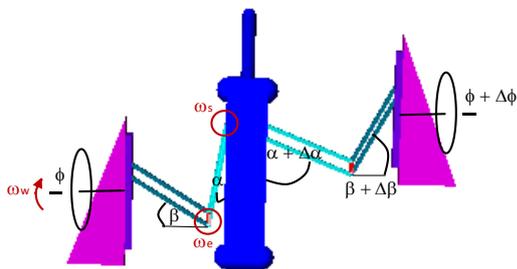


Fig 1 – Model of the DALER robot and parameters to be controlled during walking.

The optimisation is based on the cost of transport, $COT \left[\frac{J}{kg*m} \right] = \frac{Energy}{mass*distance}$.

According to previous experiments, a configuration with a fixed wingspan is chosen and the parameters for the evolution are defined by fixing some of them over the others.

The gait is then optimised for flat grounds with different friction coefficients, for several slopes and rough terrains and for different speed assigned to the robot. An example of walking on flat ground is shown by Figure 2.

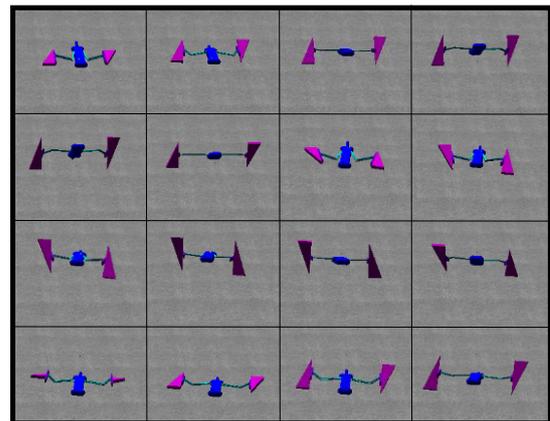


Fig 2 - Sequence of the walking of the robot on a carpet for slightly more than one rotation of wingerons.

This report finally shows that the robot can evolve different gaits depending on the terrain.

Contents

1.	Introduction.....	2
1.1.	Description of the project	2
1.2.	DALER.....	2
1.3.	Modelling of the robot.....	3
2.	Genetic Algorithms	4
2.1.	Justification of the choice of this method	4
2.2.	Design of the algorithm	4
3.	Evolution of the gait using genetic algorithm.....	6
3.1.	Choice of parameters	6
3.1.1.	Static arms	6
3.1.2.	Synchronisation between arms and flaps	6
3.2.	Fitness function of the optimisation.....	7
3.2.1.	Cost of transport.....	7
3.2.2.	Speed setting	8
4.	Results of simulations	9
4.1.	Relationship between the speed and the COT	9
4.2.	Different openings of the wings for static arms	9
4.3.	Optimisation on different grounds.....	10
4.3.1.	Flat ground with different initialisations	10
4.3.2.	Flat ground with different friction coefficients	12
4.3.3.	Slopes.....	16
4.3.4.	Rough grounds.....	19
4.4.	Optimisation for different speeds.....	20
5.	Conclusion and future work	23
	References	24

1. Introduction

1.1. Description of the project

The combination of locomotion modes allows robots to be more flexible and adaptive to their environment by using the benefits of each gait to accomplish different tasks. Walking and flying robots are particularly suited for exploration and search-and-rescue missions as they are able to perform long distance flights and to walk in complex terrains for local exploration in order to optimise the energy consumption. Conventionally, the second locomotion mode is added by incorporating another structure to an existing robot, increasing its weight and sometimes reducing the performance of its primary gait. Taking inspiration from animals, some robots with adaptive morphology are now developed. In this way, some parts of the structure are shared by multiple locomotion modes and are adjusted to adapt to the gait. Though the implementation is more complex, the number of mechanisms such as sensors and actuators is limited and each gait can be optimised.

The DALER can fly and walk using its wings. These wings are thus deployable and allow it to change its morphology to increase its efficiency. A controller is implemented to control the walking ability. The goal of this project is to improve the controller to control the deployable mechanism during the walk and thus to optimise this gait in term of cost of transport (COT). To do that, the robot is simulated on the submicroscopic realistic simulator Webots and genetic algorithms are used to evolve the synchronisation of the motors to find optimal behaviour depending on the terrain.

Subsection 1.2 of this report presents the DALER robot, which is modelled using Webots in subsection 1.3. The genetic algorithm is described in section 2. The choice of parameters for the evolutions and the fitness function used in this project are explained in section 3. Section 4 presents the results obtained for different configurations of the robot and for several terrains and section 5 concludes this report and suggest future work.

1.2. DALER

The DALER is a multi-modal bio-inspired walking and flying robot developed at the Laboratory of Intelligent System (LIS, EPFL). Its mechanics are shown in Figure 1.1.

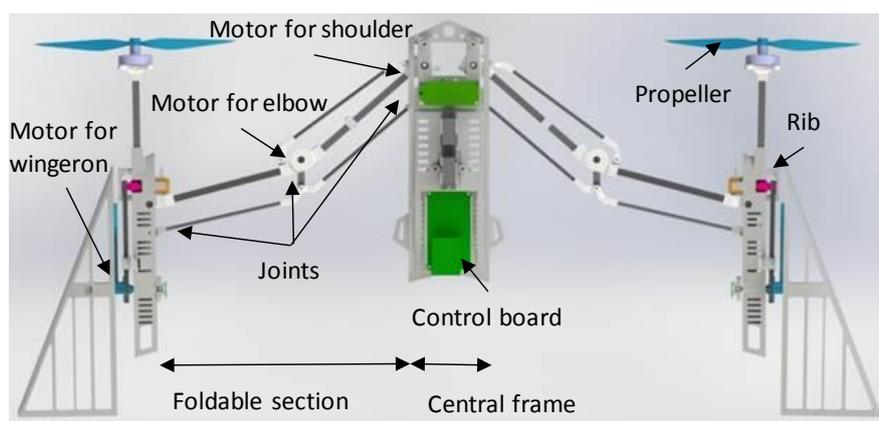


Fig. 1.1 – Mechanics and actuators of the DALER robot.

Its primary locomotion mode is flight as it is to be used for search and rescue mission and a part of the structure of the wings will be used for walking on the ground. The wingerons are the extremity of the wings and are used to control the pitch and the roll during the flight and are rotated to push the robot forward when it walks on the ground. The other part of the wings is

composed by two motorised articulations, called shoulders and elbows, and one free articulation on each side of the robot. These joints allow the wings to be deployable, as shown in Figure 1.2, and thus to adapt the morphology of the robot according to the locomotion mode. In this way, the wingspan is increased during the flight to enhance the manoeuvrability and it can be reduced during walking to improve its agility by allowing it to pass through narrow paths for example. Changing the wingspan during walking can therefore increase the efficiency of the robot in this locomotion mode.



Fig. 1.2 – The DALER robot with its maximum (left) and medium (right) wingspan.

1.3. Modelling of the robot

A model of the robot is implemented in Webots, a development environment used to model, program and simulate mobile robots. Simple shapes such as parallelepipeds, triangles and cylinders are used, as shown in Figure 1.3.

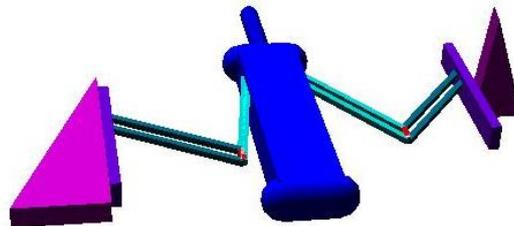


Fig. 1.3 – Model of the DALER robot on Webots.

The central frame is approximated as a parallelepiped with one capsule at each extremity of the long side. The role of these two capsules is to avoid contact between the floor and the sharp edges of the box, first because the real robot possesses chamfers and secondly because a single line of contact adds significant noise on the model. Furthermore, a third capsule is fixed on the small rear side of the central frame in order to extend it and, in this way, to prevent the rotation of the robot itself.

The different parts of the arms are modelled by parallelepipeds and are connected together and to the other components by hinge joints containing rotational motors where required. The geometry constraints the ribs to remain parallel relative to the central frame. The wingerons are modelled by triangles. No capsule is added to them because their shape makes the noise increment less significant and because friction is needed to make the robot go forward with the rotation of the flaps.

2. Genetic Algorithms

2.1. Justification of the choice of this method

As the locomotion on the ground of the robot is controlled by six motors on six articulations, the speed of each motors and the phase shifts between the different joints can be selected, so the set of parameters to be defined is significantly large. In this way, some optimisation technics are not adapted to obtain results in an acceptable amount of time.

For example, calculus-based methods search the optimum by climbing a function to set its gradient to zero. Firstly, the risk to be trapped in a local optimum is high and secondly, the function of the walk of the robot and its partial derivatives are hard to define. Another method is the enumerative search, which consists in calculating the objective function values at each point of the discretized parameters space. Doing this enumeration with the final selected parameters (see section 3.1.2) and taking a step of 0.001 (representing $\sim 0.05^\circ$) between two values of a parameters, the objective function would be evaluated $\frac{\text{span of } \omega_s}{\text{step}} * \frac{\text{span of } \varphi_s}{\text{step}} * \text{span of ratio} = \frac{3.4}{0.001} * \frac{2*\pi}{0.001} * 8 \simeq 171 \text{ mio}$ times. As a simulation takes approximately one second on a laptop, it would represent 5 years and 6 months to execute this enumerative search, which is obviously not feasible. The last “conventional” method is random search algorithm, which do sometimes not better than enumerative search and lacks efficiency.

Thereby, genetic algorithms represent a good way to optimise the parameters of the robot as, even if the global optimal is not guaranteed to be reached, they are robust in complex spaces because they search from a population of points using probabilistic transition rules so that several optimum can be evaluated in parallel. Furthermore, this method uses only an objective function, also called fitness function, making no additional knowledge necessary. Genetic algorithms are thus a simple and powerful approach adapted to complex problems like this one.

2.2. Design of the algorithm

The main steps of a simple genetic algorithm are reproduction, crossover and mutation. For the evolution of the walk of the robot, only reproduction and mutation are performed, as shown by the state machine of Figure 2.2.1.

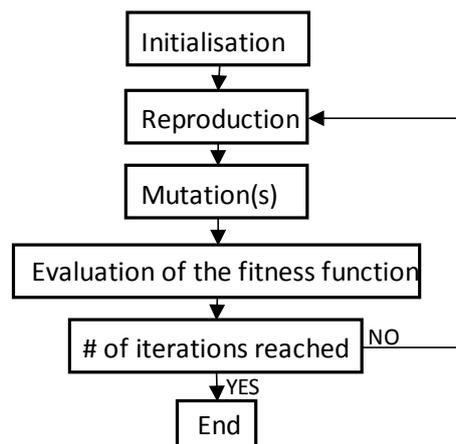


Fig. 2.2.1 – General state machine of a genetic algorithm.

The initialisation step is composed of the random selection of defined amount of sets of parameters that are simulated. For each set, the value of the fitness is calculated. During the reproduction, two sets of parameters are randomly selected, the one having the worst value of

fitness being replaced by a copy of the best one. This copy is then mutated, which means that one or more of its parameters, randomly chosen, are modified by adding a normally distributed deviation. The evolved set is then simulated and its fitness computed. These steps are repeated until the defined number of iterations is reached.

This simple approach can be improved by optimising the exploration of the space of parameters. More than one mutation is thus done at each iteration and the best of them is kept as the evolved set.

Thus, the method described above is not noise-resistant. The noise in robotic systems is significant and is principally caused by the sensors and actuators, the initials conditions and the interactions with their environment and other systems. A noise-resistant algorithm is then necessary to obtain good and robust candidate solutions and to suppress the lucky ones, which perform well only under certain conditions of noise. As the evaluation are noisy, multiple simulations are done for each set of parameters and the worst fitness value obtained is defined as the current fitness value of the set.

The retained algorithm is applied on 100 sets of parameters and is composed of 2000 iterations, each of them containing 5 mutations and 5 evaluations are performed for each set of parameters. This genetic algorithm is described by Figure 2.2.2.

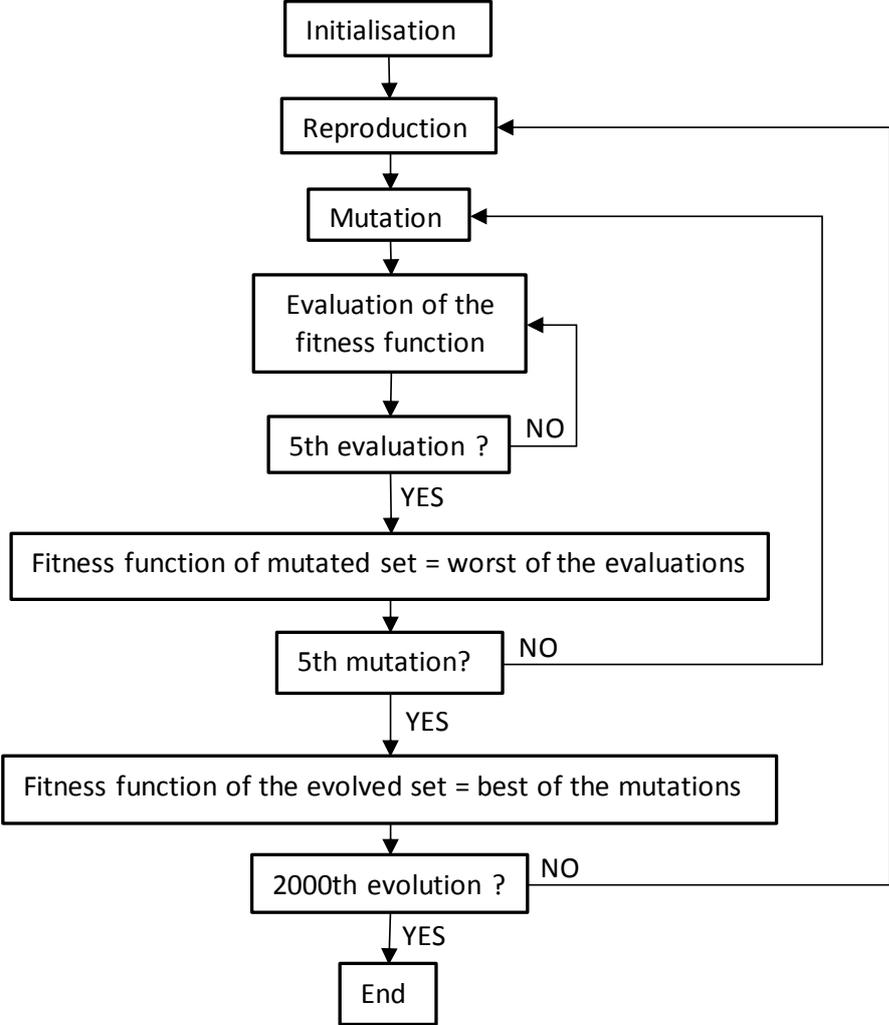


Fig. 2.2.2 – State machine of the genetic algorithm used to optimise the gait of the robot.

3. Evolution of the gait using genetic algorithm

3.1. Choice of parameters

3.1.1. Static arms

To have a comparison with the work already done on the control of the walk of the DALER [2], [3], simulations have been done with a fixed wingspan and varying the speed of the motors of wingersons. The length of the foldable section b is fixed at 155 mm so that the wings are half-opened, which seems to be the most promising solution according to previous experiments. A simple evolution has then been computed with only three parameters including two mutated as characterised in table 3.1.1.

Parameter	Min value	Max value	Mutated
Length of foldable section [mm]	155	155	No
Speed of motors of wingersons [rad/s]	0.1	3.5	Yes
Phase shift between motors of wingersons [rad]	0	2π	Yes

Table 3.1.1 – Parameters used for simple evolutions.

Several varying parameters have then been added, as described in the next section.

3.1.2. Synchronisation between arms and flaps

In order to limit the computation time of the algorithm and to increase its performance, the set of parameters which control the robot during this walk has to be well chosen. A first assumption based on observation made is that the speeds of the left and right motors of each sort of joint has to be the same so that the robot walks straight.

The parameters to be controlled during the walk process are thus:

- The speed of the left and right motors of wingersons ω_w .
- The phase shift between the motors of wingersons $\Delta\phi$.
- The speed of the left and right motors of shoulders ω_s .
- The phase shift between the motors of shoulders $\Delta\alpha$.
- The speed of the left and right motors of elbows ω_e .
- The phase shift between the motors of elbows $\Delta\beta$.
- The initial positions of three of the motors $\phi_0, \alpha_0, \beta_0$.

The different parameters are represented in Figure 5.1.2.

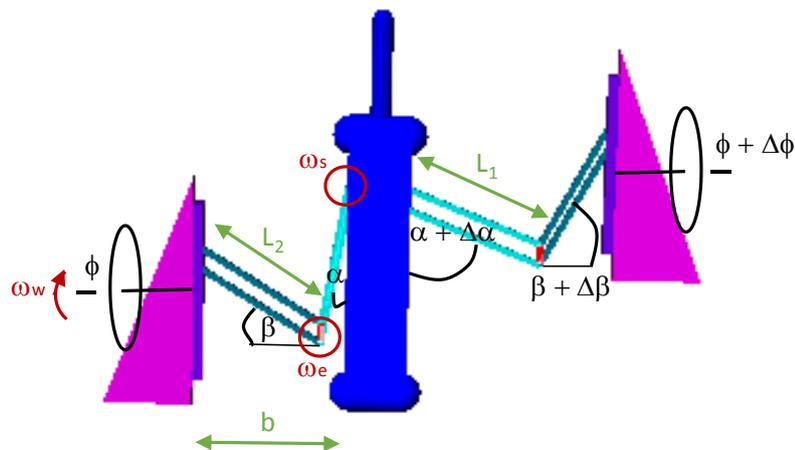


Fig. 5.1.2 – Parameters to be controlled during walking.

Furthermore, the mutation can be applied only on some parameters, so that the value of others are fixed or forced over the mutated ones. Choosing a configuration with fixed length of

foldable section allows to constraint the parameters controlling the elbows relatively to those controlling the shoulders according to equations 3.1 to 3.3.

$$\omega_e = \omega_s \frac{L_1 * \cos \alpha}{L_2 * \sin \beta} \quad (3.1)$$

$$\beta = \cos^{-1} \frac{b - L_1 * \sin \alpha}{L_2} \quad (3.2)$$

$$\beta_0 = \cos^{-1} \frac{b - L_1 * \sin \alpha_0}{L_2} \quad (3.3)$$

The speed of the motors of shoulders is determined by the speed of the motors of wingers multiplied by a constant ratio, $r \in \mathbb{N}$, according to equation 3.4.

$$\omega_s = r * \omega_w \quad (3.4)$$

In order to have the same movement for the two wings, the phase shift and the initial position of the shoulder depends on the phase and initial positions of the wingers according to equations 3.5 and 3.6.

$$\Delta\alpha = \begin{cases} \Delta\phi * \frac{\phi_{max} - \phi_{min}}{\pi}, & \text{if } \Delta\phi < \pi \\ \phi_{max} - \phi_{min} - \Delta\phi * \frac{\phi_{max} - \phi_{min}}{\pi}, & \text{else} \end{cases} \quad (3.5)$$

$$\alpha_0 = \begin{cases} \phi_0 * \frac{\phi_{max} - \phi_{min}}{\pi}, & \text{if } \phi_0 < \pi \\ \phi_{max} - \phi_{min} - \phi_0 * \frac{\phi_{max} - \phi_{min}}{\pi}, & \text{else} \end{cases} \quad (3.6)$$

The initial positions of the wingers are fixed by equations 3.7 and 3.8.

$$\phi_{left,0} = 0 \quad (3.7)$$

$$\phi_{right,0} = \Delta\phi \quad (3.8)$$

The speed and phase shift of wingers and the ratio between the speeds of the motors of wingers and shoulders are thus mutated parameters. The more complex evolution is then done with eight main parameters, without the initial positions of the motors, as characterised in table 3.1.3.

Parameter	Min value	Max value	Mutated
Length of foldable section [mm]	155	155	No
Speed of motors of wingers [rad/s]	0.1	3.5	Yes
Phase shift between motors of wingers [rad]	0	2π	Yes
Ratio r	0	8	Yes
Speed of motors of shoulders [rad/s]	0	8	No
Phase shift between motors of shoulders [rad]	0	2π	No
Speed of motors of elbows [rad/s]	0	8	No
Phase shift between motors of elbows [rad]	0	2π	No

Table 3.1.3 – Parameters used for complex evolutions.

3.2. Fitness function of the optimisation

3.2.1. Cost of transport

The performance of each set of parameters is evaluated through the value of its fitness function. The fitness function minimized is the cost of transport (COT) of the robot, given by equations 3.9 and 3.10. The distance used is the distance travelled along the main axis of the robot.

$$COT \left[\frac{J}{kg*m} \right] = \frac{Energy}{mass*distance} \quad (3.9)$$

$$Energy [J] = \sum_{i=1}^6 \sum_{t=0}^{t_{end}} \frac{1}{\eta_{i,\omega(t)}} * \tau_{i,t} * \delta\varphi_{i,t,t-\Delta t} \quad (3.10)$$

Where $i = [1; 6]$ is the index of the motors, t_{end} is the time of the walk, $\tau_{i,t}$ is the torque of motor i at time t , $\delta\varphi_{i,t,t-\Delta t}$ is the difference of angle of motor i between time $t-\Delta t$ and t and $\eta_{i,\omega(t)}$ is the efficiency of motor i at speed $\omega(t)$, given by equation 3.11. The efficiency in function of the ratio between the speed and the nominal speed is shown on figure 3.1.4.

$$\eta_{i,\omega(t)} = \eta_{max,i} * \left(- \left(\frac{\omega_i(t)}{\omega_{nominal,i}} \right)^2 + 2 * \frac{\omega_i(t)}{\omega_{nominal,i}} \right) \quad (3.11)$$

Where $\eta_{max,i}$ and $\omega_{nominal,i}$ are respectively the best motor efficiency and the nominal speed of motor i . These parameters are given in table 3.1.5.

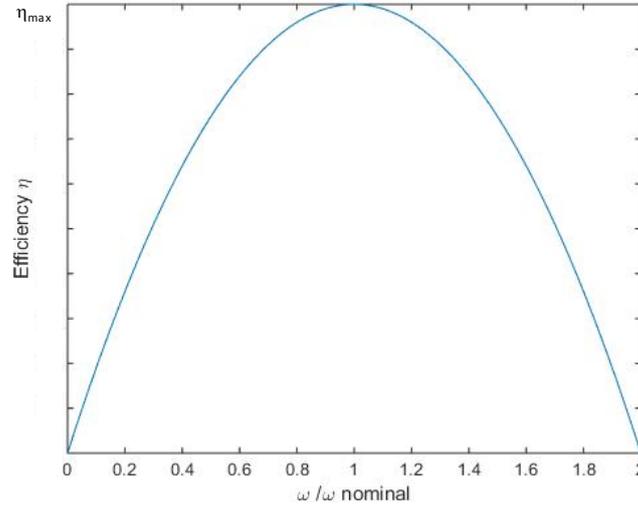


Fig. 3.1.4 – Curve of the efficiency η of a motor in function of its rotation speed ω .

	Nominal velocity $\omega_{nominal}$	Efficiency η_{max}
Motors of wingerons	2.5 rad/s	1
Servomotors of shoulders and elbows	6 rad/s	1

Table 3.1.5 – Parameters of the motors.

3.2.2. Speed setting

Another fitness is needed to evaluate the best set of parameters to set the robot at a given speed v_s . This function is given by equation 3.12.

$$Fitness\ function = COT * |v_s - v| \quad (3.12)$$

Where v is the mean velocity of the robot.

4. Results of simulations

4.1. Relationship between the speed and the COT

The COT of the robot has been measured in Webots for three different openings of the wings. The simulations were performed on a parquet, with the friction coefficients of the first column of table 4.3.3, and five complete rotation of the wingerons were performed for each of them with no phase and no rotation of the other motors. Figure 4.1.1 shows the COT with closed, half-opened and opened wings as a function of the speed of the robot. The left graph was obtained by setting the speed of the motors of the wingerons from 0.1 to 3.5 rad/s with steps of 0.1 rad/s and the right graph was obtained by selecting 100 random speeds between 0.1 and 3.5 rad/s.

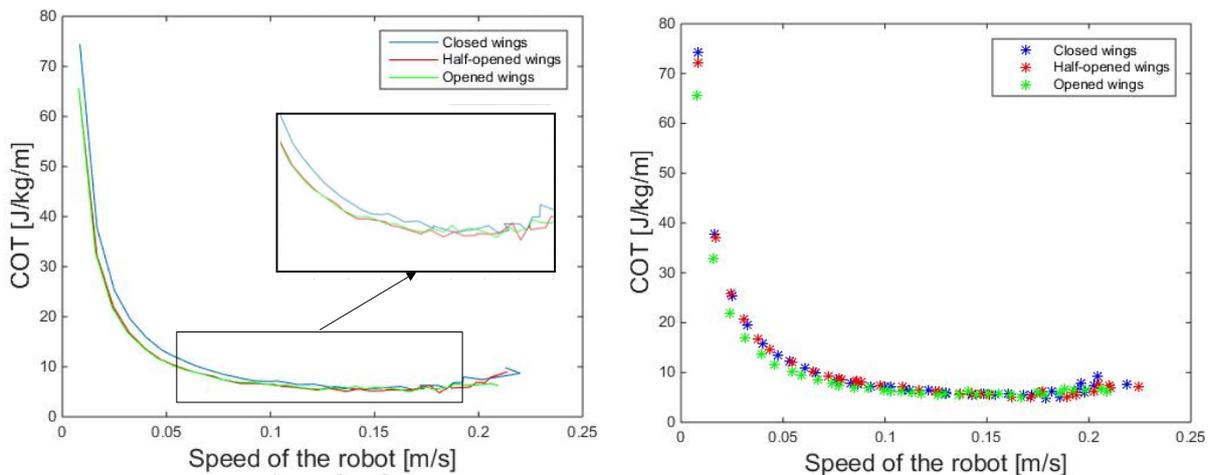


Fig. 4.1.1 – COT of the robot walking for three different openings of the wings (closed, half-opened and opened) calculated at each point of the discretized parameters space (left) and obtained with random initialisation of 100 sets of parameters (right).

The results are similar for the two graphs. The curves of the three configurations have the same shape: the COT is high for small speeds, then minimal for speeds that are around 60 and 80% of the maximal speed and it increases for high speeds. The test on the DALER robot made by [1] showed a lower COT respectively for half-closed wings at low speed and for closed wings at high speed. This tendency is not very pronounced in the results of the present simulations as the three curves are close to each other. This can be explained by the fact that the impact of the friction is less important in the simulation due to the model that is simpler than the real robot.

4.2. Different openings of the wings for static arms

The genetic algorithm with the parameters of section 3.1.1 was applied on the walking of the robot for three different openings of the wings (closed, half-opened and opened). The simulations were performed with the same conditions described in 4.1. Figure 4.2.1 shows the results obtained: the left graphs present the COT as a function of the speed of the robot for each of the 100 sets of parameters before and after the evolution, the middle graphs show the best COT of each iteration and the right graphs present the mean COT of the 100 sets of parameters at each iteration.

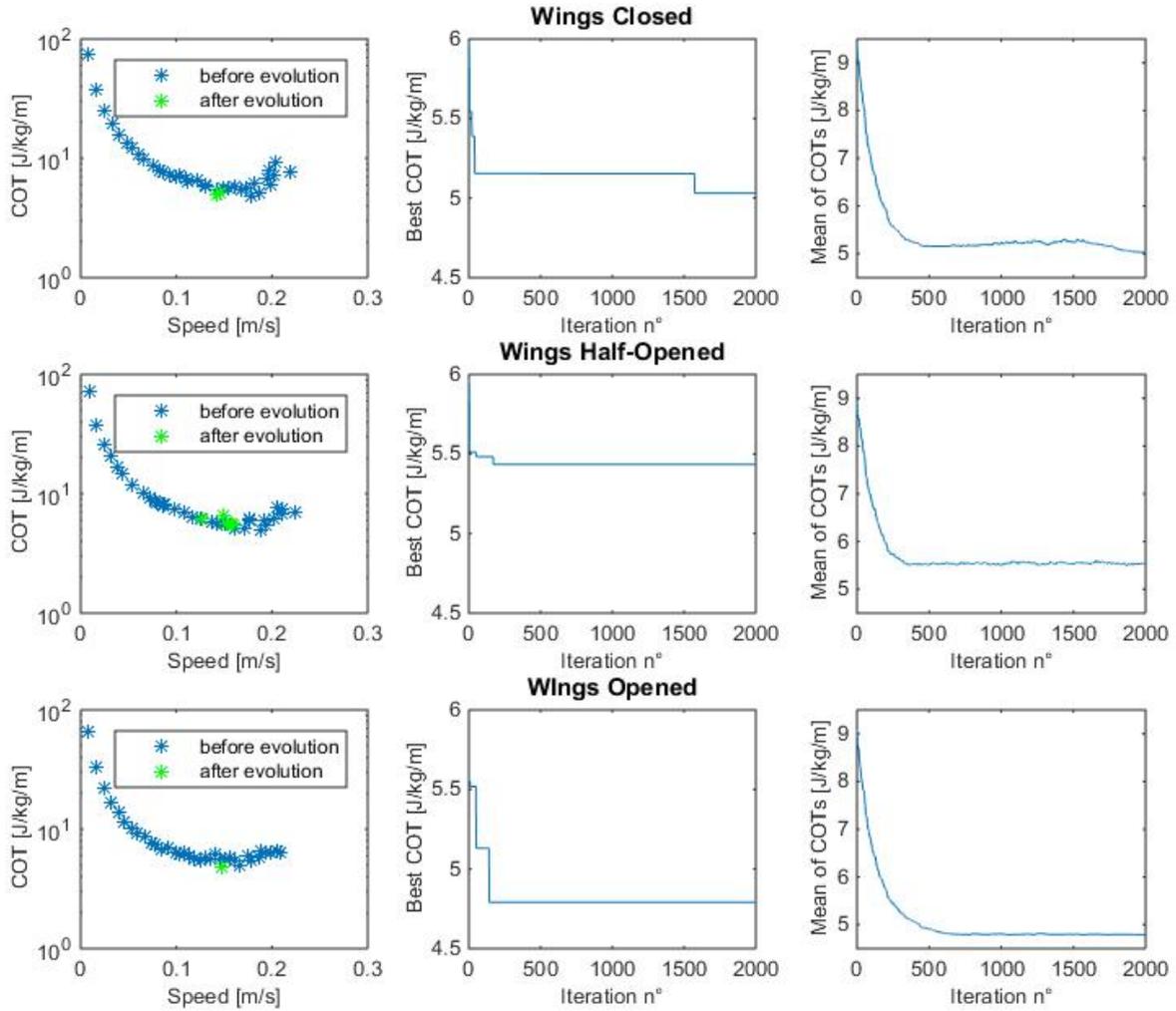


Fig. 4.2.1 – Results obtained for the evolution of the robot walking for three openings of the wings (closed, half-opened and opened). From left to right: COT obtained before and after evolution, best COT of each iteration and mean of the COTs of all the sets of parameters for each iteration.

In this case, each evolution converge to a few sets of parameters. For each opening, the lower COT for the walking is obtained at a speed of the robot equal to approximately 75% of the maximum speed reached by the random initialisation of the parameters. Moreover, though the initialisation picks different sets of parameters, the evolved sets tend to a single set for each openings. In fact, for closed, half-opened and opened wings, the optimised speed of the motors of wingerons are respectively 2, 2.32 and 2.34 rad/s. Another interesting observation is that, for each case, the best COT is obtained without any phase shift between the wingerons. Furthermore, the COT is similar for all the openings.

4.3. Optimisation on different grounds

4.3.1. Flat ground with different initialisations

The goal of this sub-section is to verify that the genetic algorithm provides valid results with limited influence of the initialisation with random parameters. The algorithm with the parameters of section 3.1.2 was applied on the walking of the robot for five different initialisations. According to the function `ran2` of [5], the variable `idum` determines the sequence of random numbers selected, so that five different `idum` define five different initialisations.

Again, the simulations were performed on a parquet, with the friction coefficients of the first column of table 4.3.3, and five complete rotation of the wingerons were performed for each of them.

Figure 4.3.1 shows the COT as a function of the speed of the robot before and after the evolution, the best COT obtained for each iteration of the algorithm and the mean COT at each iteration for initialisations with idum equal to 2, 4, 6, -1 and -4. Table 4.3.2 gives the parameters and the COT corresponding to the best result obtained after the evolution for each initialisation.

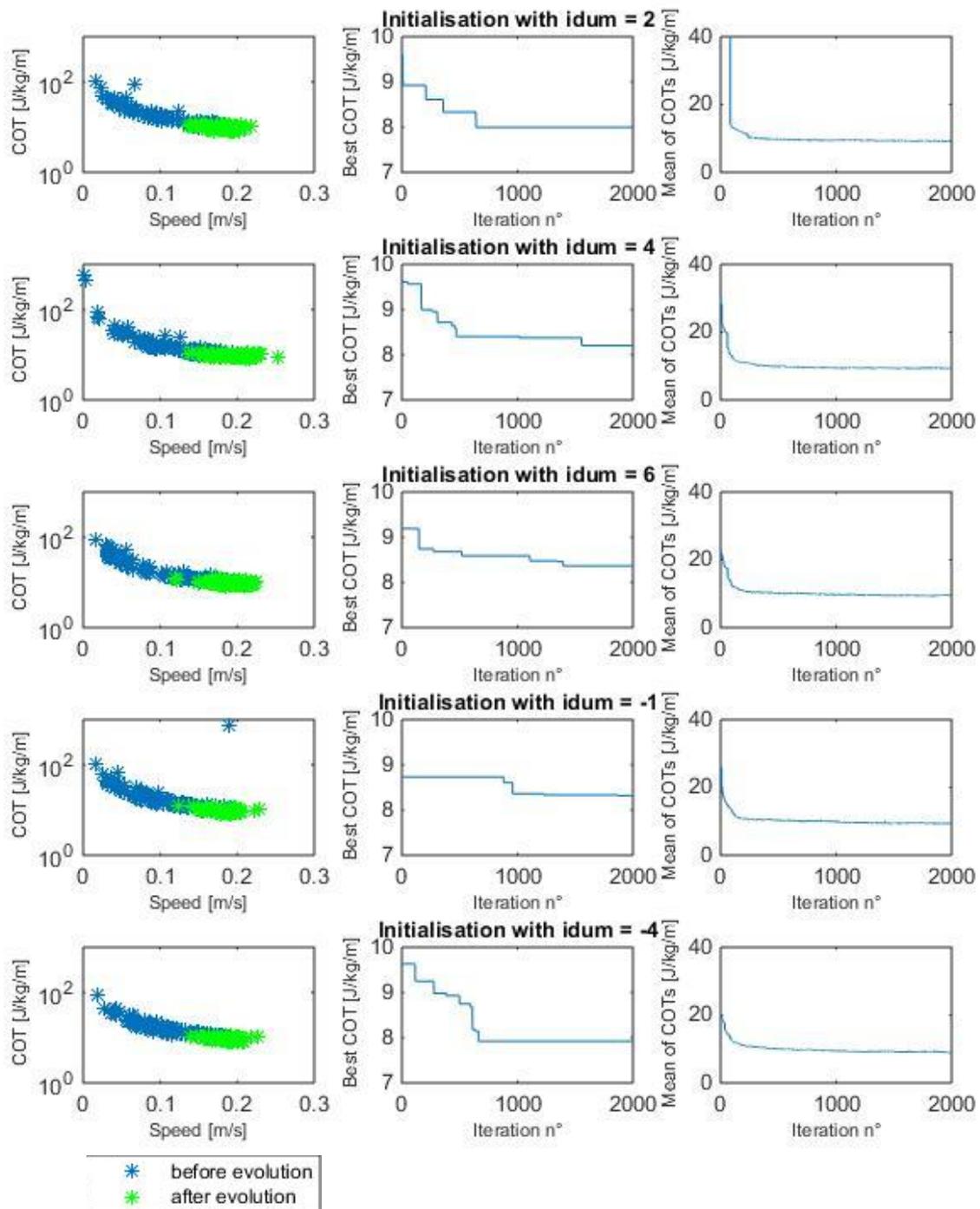


Fig. 4.3.1 – Results obtained for the evolution of the robot walking for 5 initialisations. From left to right: COT obtained before and after evolution, best COT of each iteration and mean of the COTs of all the sets of parameters for each iteration.

Idum	ω_w [rad/s]	$\Delta\phi$ [rad]	r	COT [J/kg*m]
2	2.96	3.12	2	7.99
4	3.04	3.13	2	8.02
6	2.99	3.11	2	8.37
-2	2.96	3.13	2	7.99
-4	2.63	3.13	3	7.92

Table 4.3.2 – Best COTs and corresponding sets of parameters obtained after the evolution for each idum.

The best COT and the maximal speed of the robot obtained are approximately equal to 8 J/kg*m and 0.23 m/s in all the cases, which corresponds to a diminution of 10 to 20% of the initial best COT and to an increase of 10 to 15% of the speed of the robot.

Figure 4.3.3 shows the COT as a function of the sets of parameters after the evolution. The darker the points are, the lower is the COT of the corresponding set of parameters. All the graphs have the same tend: the best COTs are obtained for $\omega_w \in [2.3; 3.5]$ rad/s, ϕ_w equal to π and r equal to 2 or 3.

The assumption of this sub-section is so validated and all next simulations are performed using a single initialisation.

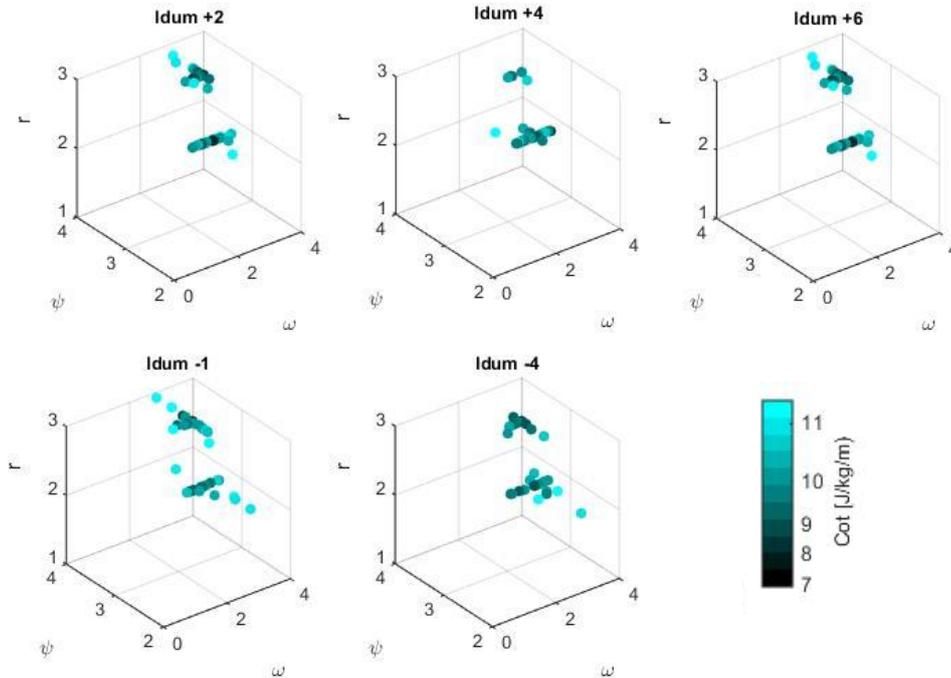


Fig. 4.3.3 – COT obtained for the sets of parameters found by the evolution of the robot walking for 5 initialisations. The parameters are ω , ϕ and r , which correspond respectively to the speed of the motors of wingerons, the phase shift between these motors and the ratio between the speed of the motors of shoulders and wingerons.

4.3.2. Flat ground with different friction coefficients

The genetic algorithm with the parameters of section 3.1.2 was applied for simulations of the walking of the robot on four different flat grounds to analyse the influence of the surface on the parameters. These surfaces corresponds to a parquet, a carpet, a road and grass and are characterised by the friction coefficients between the surface and the robot body, respectively the robot wings given by table 4.3.4. Five rotations of wingerons were performed for each simulation.

Figure 4.3.5 shows the COT as a function of the speed of the robot before and after the evolution, the best COT obtained for each iteration of the algorithm and the mean COT at each iteration for walking on a parquet, on a carpet, on the road and on the grass. Table 4.3.6 gives the parameters and the COT corresponding to the best result obtained after the evolution for each ground.

	Parquet	Carpet	Road	Grass
Robot center μ_a	0.3	0.4	0.5	0.5
Wings μ_b	0.45	1.8	1.7	1.6

Table 4.3.4 – Friction coefficients of different grounds.

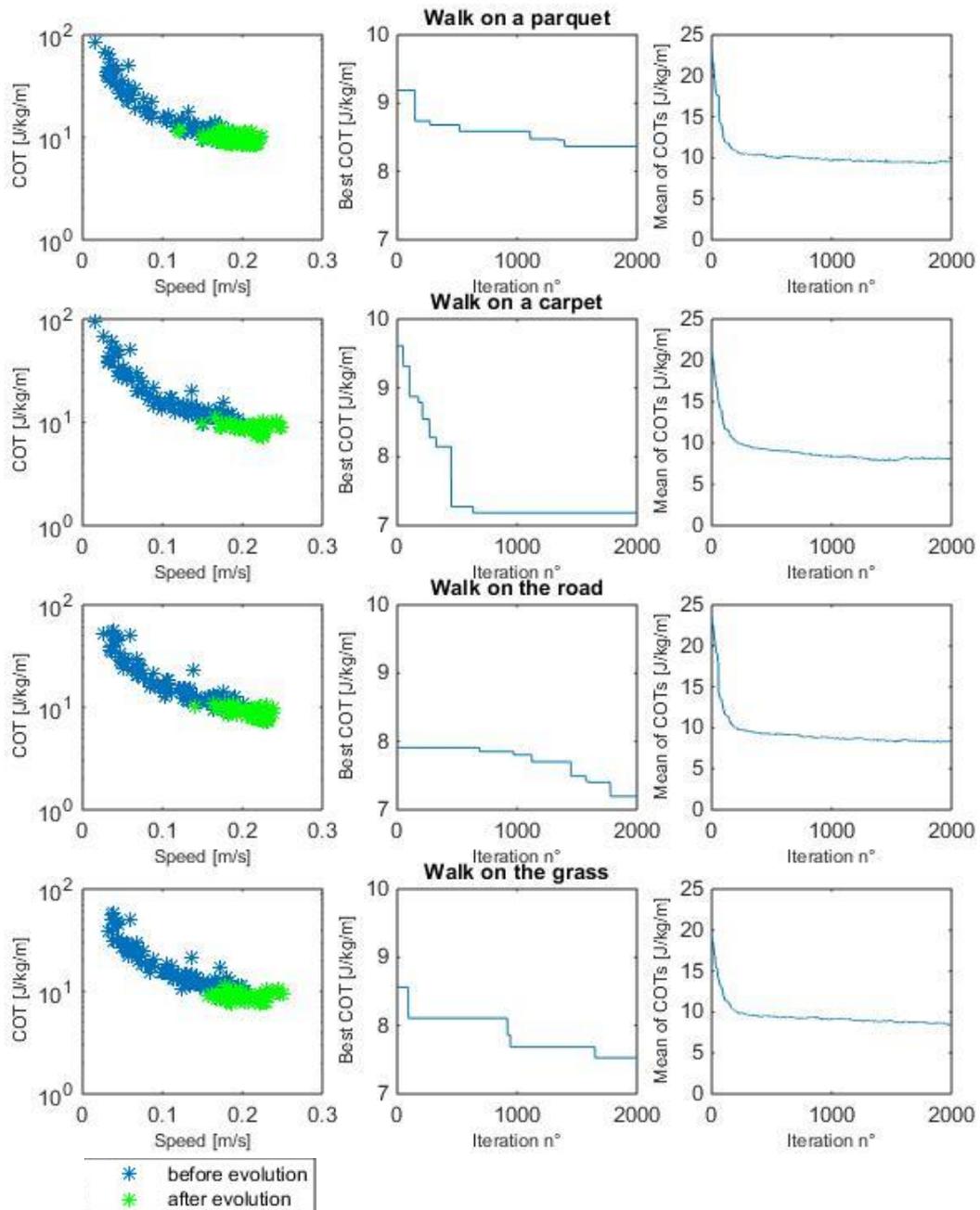


Fig. 4.3.5 – Results obtained for the evolution of the robot walking on different grounds with friction coefficients of table 4.3.4. From left to right: COT obtained before and after evolution, best COT of each iteration and mean of the COTs of all the sets of parameters for each iteration.

Surface	ω_w [rad/s]	$\Delta\phi$ [rad]	r	COT [J/kg/m]
Parquet	2.99	3.11	2	8.37
Carpet	2.94	3.14	2	7.18
Road	2.95	3.12	2	7.53
Grass	2.95	3.13	2	7.19

Table 4.3.6 - Best COTs and corresponding sets of parameters obtained after the evolution for each surface.

The robot is thus more efficient on surfaces with a high friction coefficient μ_b as on a carpet and on a road. In contrast, the COT is higher on slippery surfaces like a parquet since it is slipping a consequent portion of each step. Moreover, the speed reached are lower for slippery surface.

Figure 4.3.7 shows the COT as a function of the sets of parameters after the evolution. The darker the points are, the lower the COT of the corresponding set of parameters is. For all the surfaces, the best COTs are obtained for $\omega_w \in [2.3; 3.5]$ rad/s, ϕ_w close to π and r equal to 2 or 3. An interesting observation is that, the COTs obtained for the walking on a slippery surface like the parquet, are better for smaller speeds of motors of wingerons, typically $\omega_w \in [2.3; 2.9]$ rad/s. This can be explained by the fact that the portion of each step during which the robot is slipping is reduced when the speed is lower, so that the COT is less important.

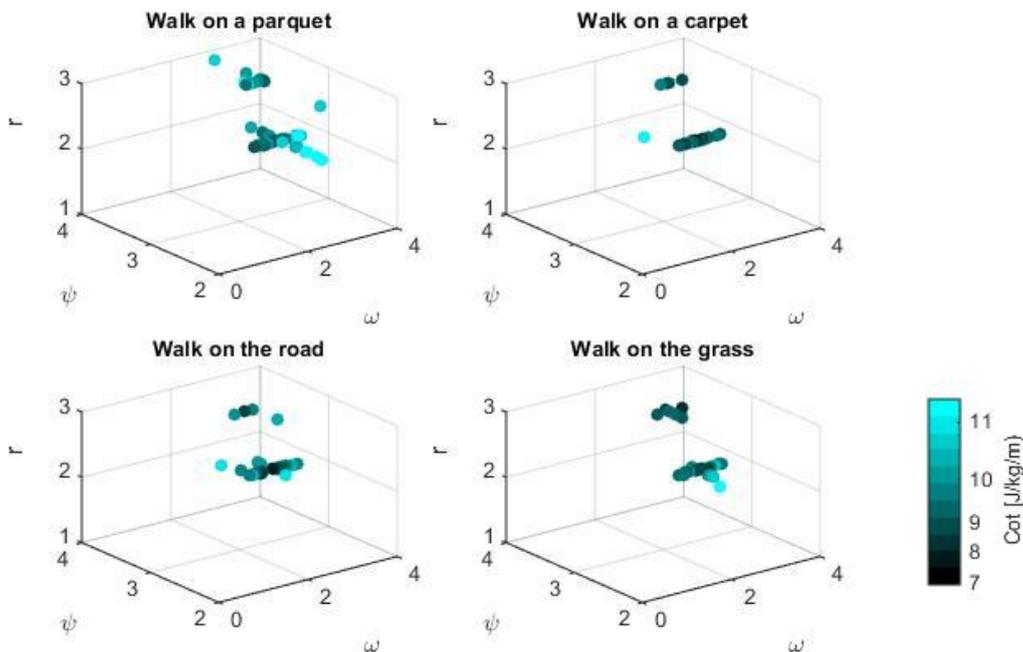


Fig. 4.3.7 – COT obtained for the sets of parameters found by the evolution of the robot walking on different grounds with friction coefficients of table 4.3.4. The parameters are ω , ϕ and r , which correspond respectively to the speed of the motors of wingerons, the phase between these motors and the ratio between the speed of the motors of shoulders and wingerons

Figure 4.3.8 shows an example of the walking of the robot on a carpet for slightly more than one rotation of the wingerons. The best parameters found by the evolution are $\omega_w = 2.94$ rad/s, $\phi_w = \pi$ and $r = 2$.

Figure 4.3.9 shows the positions of the joints over time for the walking of the robot with the same parameters for five rotations of the wingerons. As the wingerons turn continuously, their

positions jump from π to $-\pi$ at each rotation, which is the cause of the discontinuities on the graph. An interesting observation is that the shoulders and elbows move symmetrically due to the fact that $\phi_w = \pi$. As observed in previous sections, the phase shift between wingerons of the evolved sets of parameters is equal to 0 or π to have a low COT. The assumption can be made that, to achieve an efficient walking, the motors of shoulders and elbows need to be synchronised in order that they move symmetrically and keep the robot going straight.

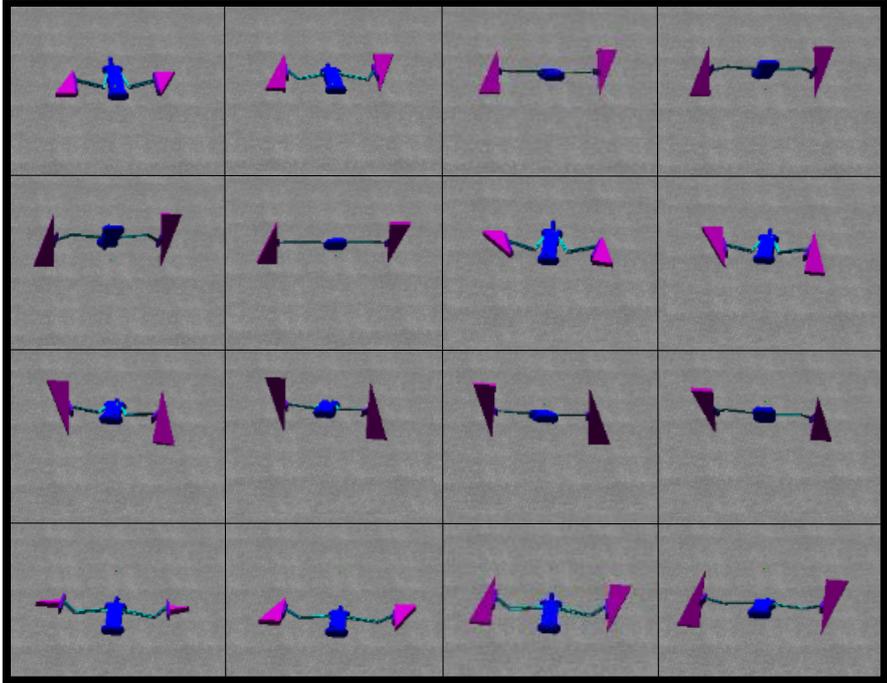


Fig 4.3.8 – Walking sequence of the robot on a carpet for slightly more than one rotation of wingerons.

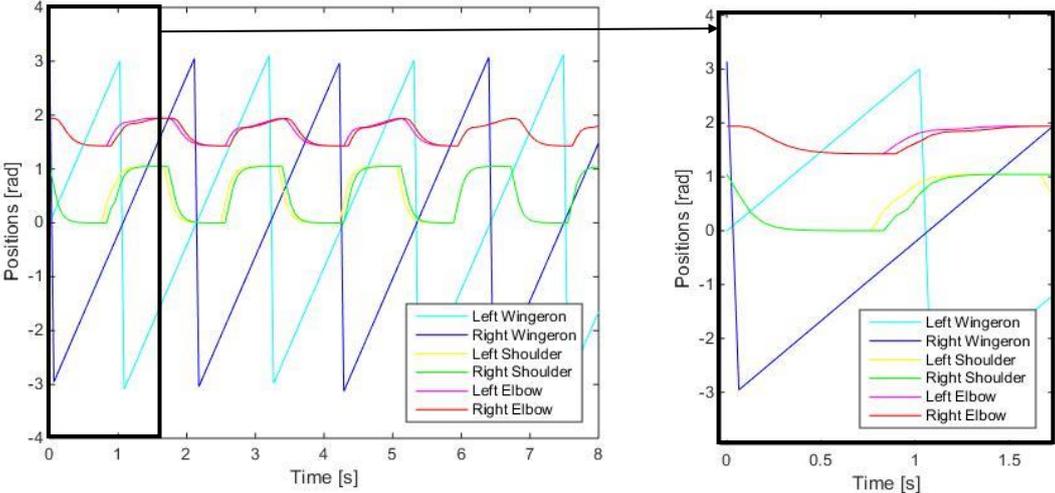


Fig. 4.3.9 – Positions of the joints as function of the time for a walking on a carpet and five rotations of the wingerons.

4.3.3. Slopes

The genetic algorithm with the parameters of section 3.1.2 was applied on simulations of the walking of the robot on slopes tilted 5° , 10° up and 5° , 10° , 15° down. The simulations were performed on a parquet, with the friction coefficients of the first column of table 4.3.4, and five rotations of wingersons were performed for each simulation.

Figure 4.3.10 shows the COT as a function of the speed of the robot before and after the evolution, the best COT obtained for each iteration of the algorithm and the mean COT at each iteration for walking on different slopes up and down. Table 4.3.11 gives the parameters and the COT corresponding to the best result obtained after the evolution for each slope.

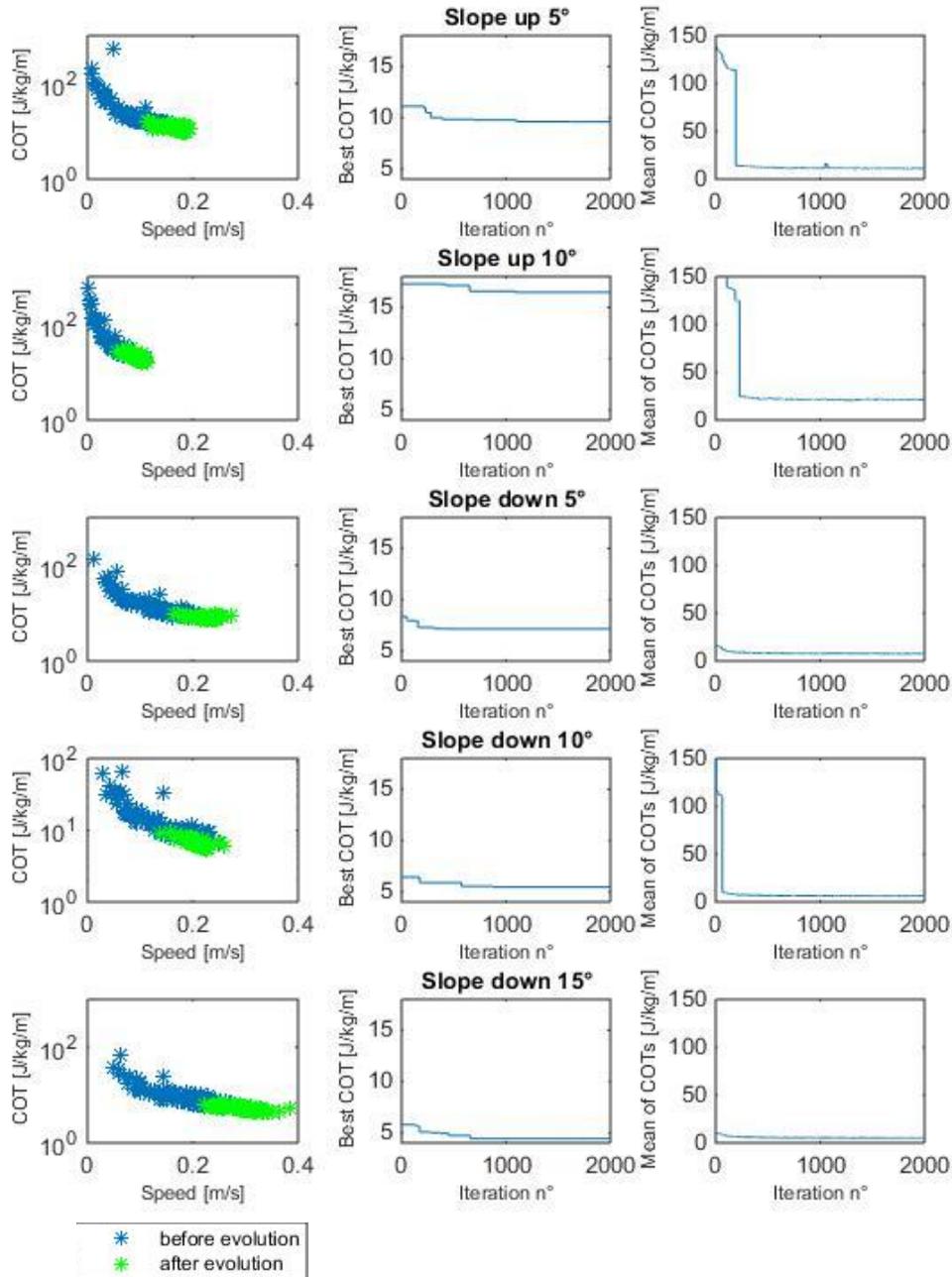


Fig. 4.3.10 – Results obtained for the evolution of the robot walking on different slopes up and down. From left to right: COT obtained before and after evolution, best COT of each iteration and mean of the COTs of all the sets of parameters for each iteration.

Slope	ω_w [rad/s]	$\Delta\phi$ [rad]	r	COT [J/kg/m]
up 5°	2.94	3.14	2	9.59
up 10°	2.32	1.96	1	16.47
down 5°	3.02	3.14	2	7.13
down 10°	2.01	3.13	3	5.45
down 15°	2.95	3.08	2	4.40

Table 4.3.11 - Best COTs and corresponding sets of parameters obtained after the evolution for each slope.

As expected, the best COT increases and the maximal speed decreases when the slope up becomes more inclined. The best COT and maximal speed obtained for a 10° tilted slope up are respectively multiplied and divided by two compared with a flat floor. The inverse phenomena happens logically for slopes down, that is the best COT decreases and the maximal speed increases. The best COT and maximal speed obtained for a 15° tilted slope down are respectively divided and multiplied by two compared with a flat floor.

Figure 4.3.12 shows the COT as a function of the sets of parameters after the evolution. The parameters are ω , ϕ and r , which correspond respectively to the speed of the motors of wingerons, the phase shift between these motors and the ratio between the speed of the motors of shoulders and wingerons. The darker the points are, the lower is the COT of the corresponding set of parameters. For slopes down and for slightly inclined slope up (5°), the best COTs are obtained for $\omega_w \in [2.3; 3.5]$ rad/s, ϕ_w close to π and r equal to 2 or 3, as for flat grounds. However, for more tilted slope up (10°), the best COTs are obtained for smaller ω_w , as $\omega_w \in [1.7; 2.8]$, $\phi_w \in [\pi/2; 3\pi/2]$ and r often equal to 1.

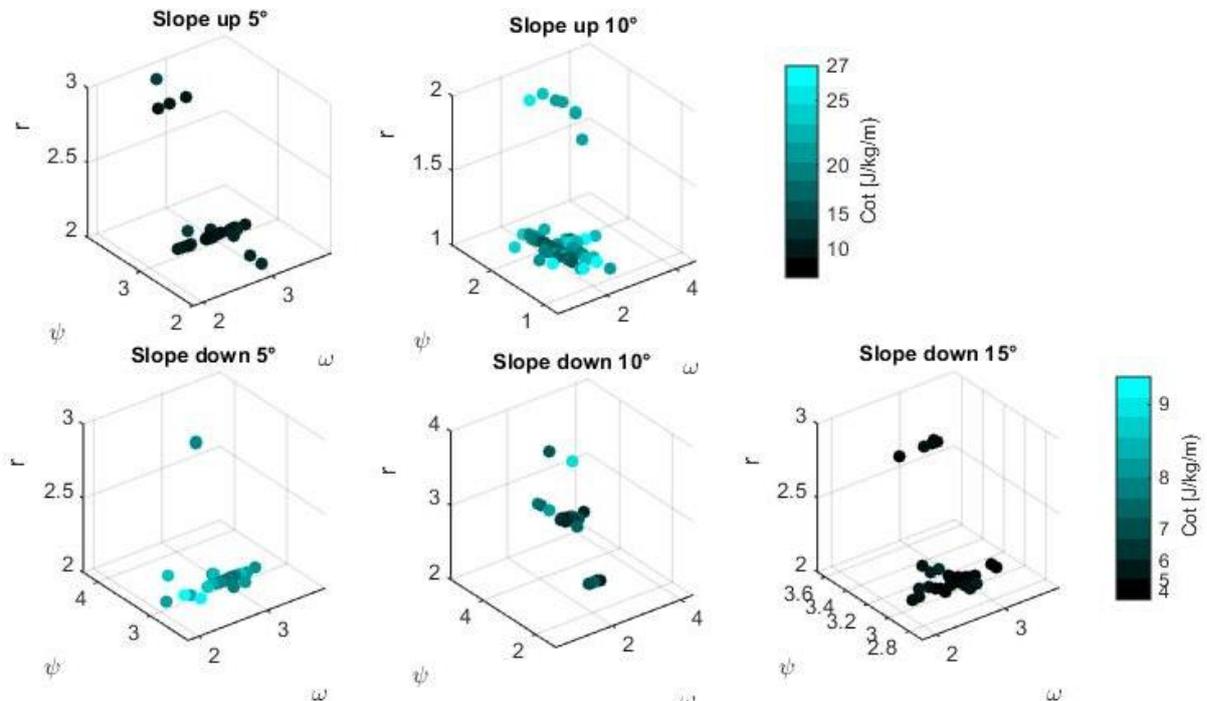


Fig. 4.3.12 – COT obtained for the sets of parameters found by the evolution of the robot walking on different slopes up and down

Figure 4.3.13 shows an example of the walking of the robot on a slope up tilted 10° for one rotation of the wingerons. The best parameters found by the evolution are $\omega_w = 2.33$ rad/s, ϕ_w

$= 1.96$ and $r = 1$. Moreover, Figure 6.3.14 shows the positions of the joints over time for the walking of the robot with the same parameters for three rotations of the wingerons.

This sequence is an example where the phase of wingerons is different from 0 or π so that the shoulders and elbows do not move symmetrically. In the case of an inclined slope, it allows the robot to use one wingeron after the other to “grip” the ground and to “climb” the slope, as a climber would do with two ice axes. This gait allows it to compensate the slipperiness of the slope.

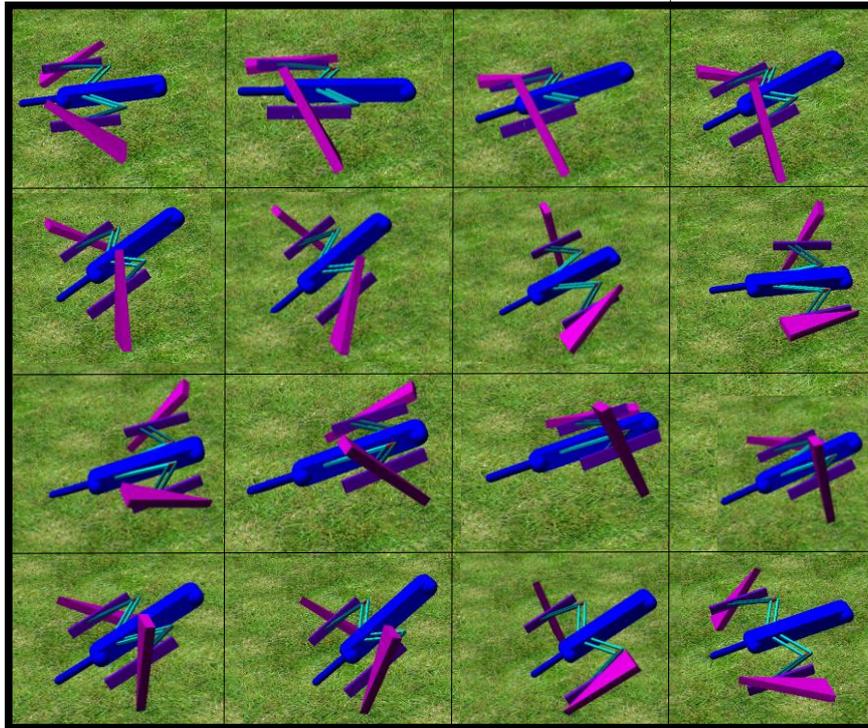


Fig. 4.3.13 – Sequence of the walking of the robot on a slope up tilted 10° for one rotation of wingerons.

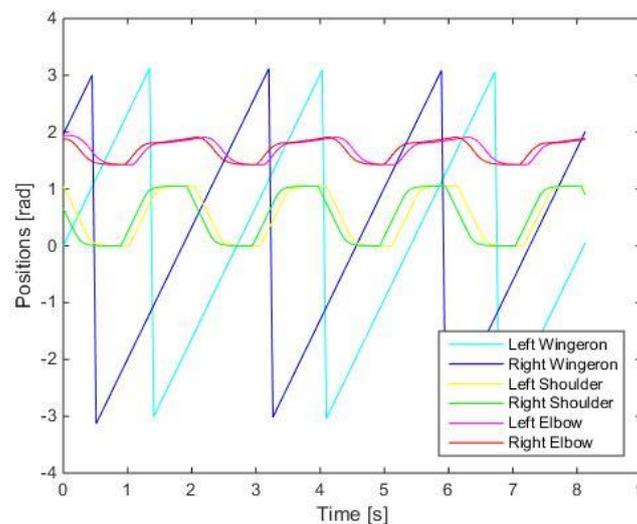


Fig. 4.3.14 - Positions of the joints as function of the time for walking on a slope up tilted 10° and three rotations of the wingerons.

4.3.4. Rough grounds

The genetic algorithm with the parameters of section 3.1.2 was applied on simulations of the walking of the robot on two rough terrains shown by Figure 4.3.15. Rough terrain 1 has more difference in altitude than rough terrain 2, but it is also more regularly disposed as the change of height are slow. The simulations were performed on a parquet, with the friction coefficients of the first column of table 4.3.4, and five rotations of wingerons were performed for each simulation.

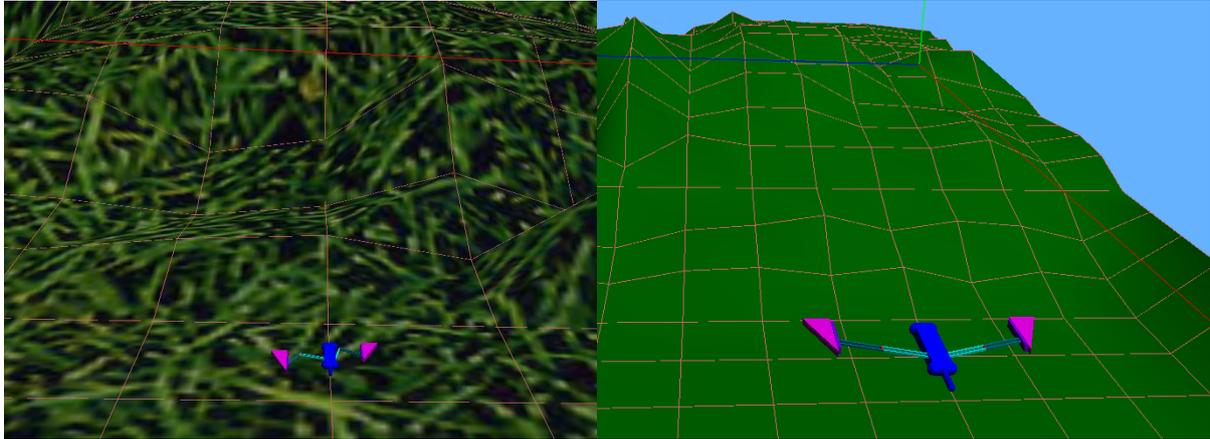


Fig. 4.3.15 – Rough terrains 1 (left) and 2 (right).

Figure 4.3.16 shows the COT as a function of the speed of the robot before and after the evolution and the best COT obtained for each iteration of the algorithm for walking on the two rough terrains. Table 4.3.17 gives the parameters and the COT corresponding to the best result obtained after the evolution for each ground. The performance is better on rough terrain 1, which altitude changes slowly.

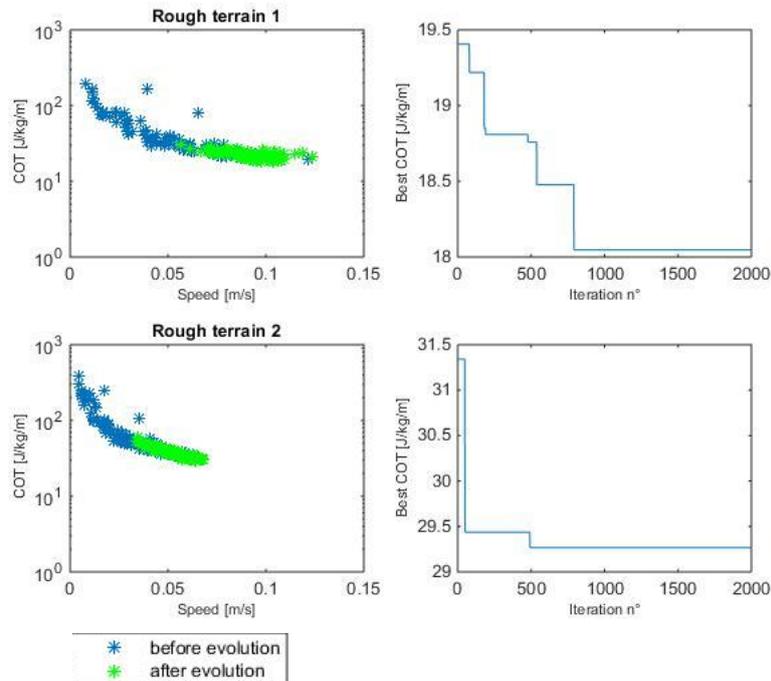


Fig. 4.3.16 – Results obtained for the evolution of the robot walking on different rough grounds of Figure 4.3.15. From left to right: COT obtained before and after evolution and best COT of each iteration.

Rough terrain	ω_w [rad/s]	$\Delta\phi$ [rad]	r	COT [J/kg/m]
1	2.27	1.84	1	18.05
2	2.81	1.51	2	29.26

Table 4.3.17 - Best COTs and corresponding sets of parameters obtained after the evolution for each rough ground.

Figure 4.3.18 shows the COT as a function of the sets of parameters after the evolution. The darker the points are, the lower the COT of the corresponding set of parameters is. As for inclined slopes up, ω_w tends to be smaller, ϕ_w is not equal to 0 or π and r often equal to 1. Once again, this allows the robot to “grip” the ground to compensate the slipperiness of the ground.

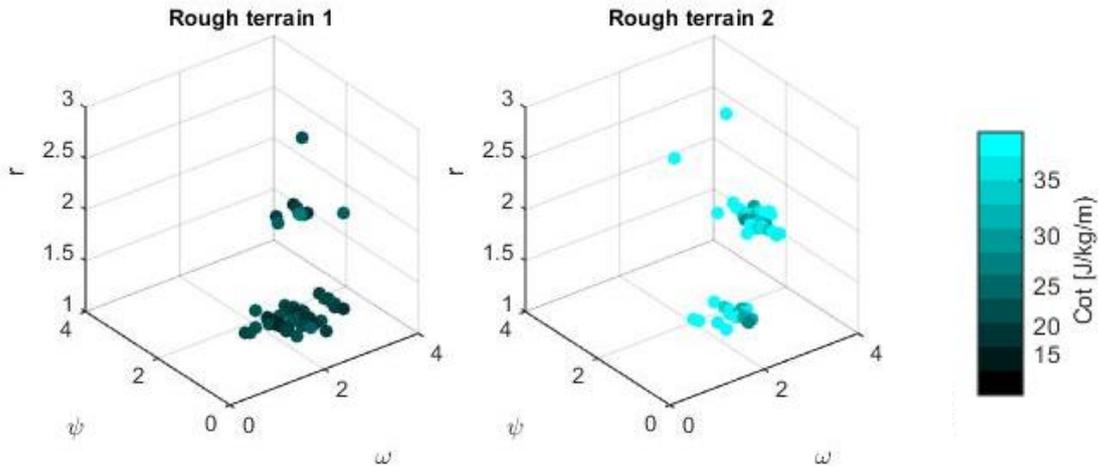


Fig. 4.3.18 – COT obtained for the sets of parameters found by the evolution of the robot walking on different rough grounds of Figure 4.3.15.

4.4. Optimisation for different speeds

The goal of this sub-section is to optimise the set of parameters needed to make the robot walks at a given speed. The fitness function used is then the one described in section 3.2.2 with the parameters of section 3.1.2. The simulations were performed on a parquet, with the friction coefficients of the first column of table 6.3.3, and five rotations of wingersons were performed for each simulation. The given speed are 0.05, 0.1, 0.15, 0.2 and 0.25 m/s.

Figures 4.4.1 shows the COT as a function of the speed of the robot before and after the evolution, the best COT obtained for each iteration of the algorithm and the mean COT at each iteration for walking on flat ground. Table 4.4.2 gives the parameters and the COT corresponding to the best result obtained after the evolution for each ground.

The evolved sets of parameters do not really differ from the sets initialised. The algorithm just favours the sets of the beginning that correspond to the given speed. The only exception is for the speed of 0.25 m/s, for which the algorithm provides lower COT and higher maximal speed. Figure 4.4.3 shows the COT as a function of the sets of parameters after the evolution. The darker the points are, the lower is the COT of the corresponding set of parameters. For small given speeds such as 0.05 and 0.1 m/s, ω_w is enough small not to go faster than the speed wanted, ϕ_w does not seem to have an important influence as r , which is sometimes small, sometimes high without any visible reason. For a given speed of 0.15 m/s, ω_w and ϕ_w are respectively approximately equal to 2 rad/s and π rad and r is equal to 1, 2 or 3, so that is does not seem to have a significative influence. For higher speeds, ϕ_w does not seem to be influent.

The speeds are obtained by setting the adapted speed of the motors of wingerons and a ratio r of 2.

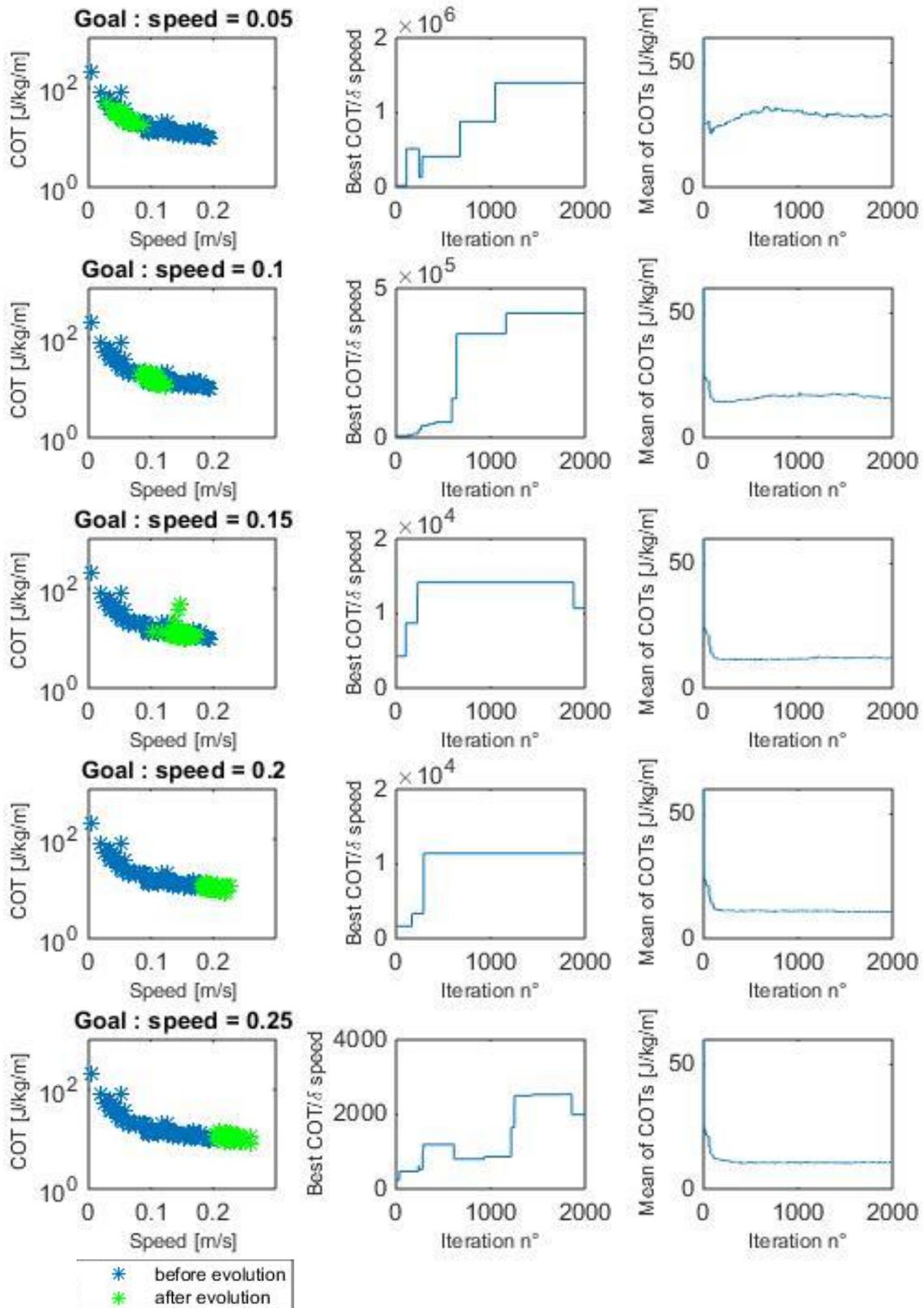


Fig. 4.4.1 – Results obtained for the evolution of the robot walking at different given speeds. From left to right: COT obtained before and after evolution, best COT of each iteration and mean of the COTs of all the sets of parameters for each iteration

Goal speed [m/s]	ω_w [rad/s]	$\Delta\phi$ [rad]	r	COT [J/kg/m]
0.05	0.77	4.79	3	27.96
0.1	1.43	2.47	1	18.75
0.15	2.27	3.20	1	11.12
0.2	3.06	3.92	1	11.68
0.25	3.5	3.24	2	9.09

Table 4.4.2 - Best COTs, speed and corresponding sets of parameters obtained after the evolution for each speed setted.

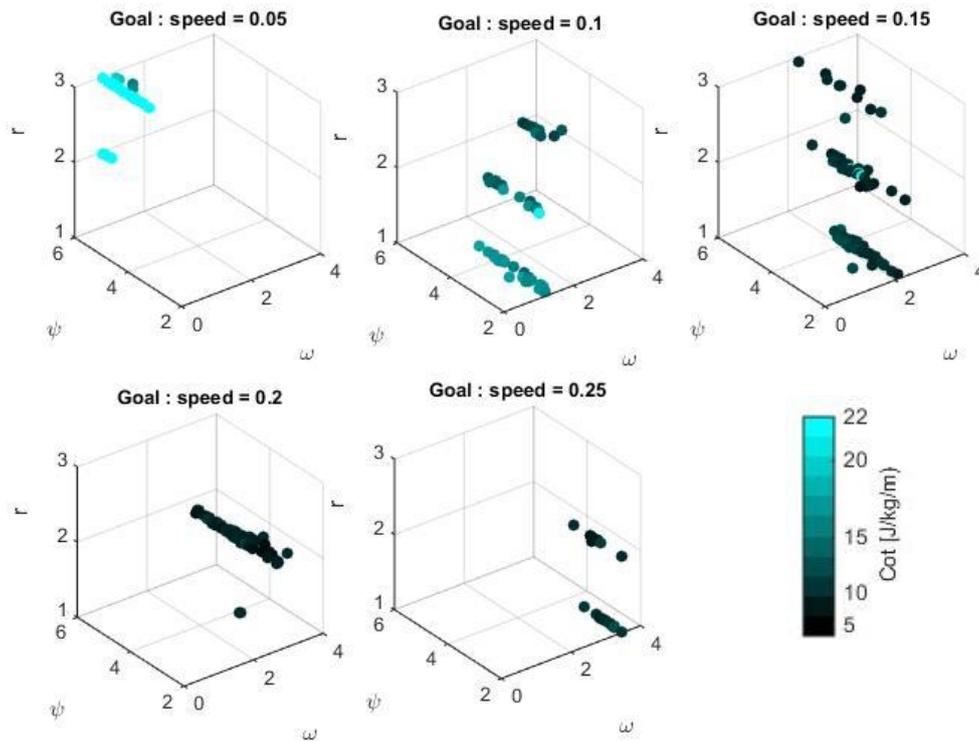


Fig. 4.4.3 – COT obtained for the sets of parameters found by the evolution of the robot walking at different speeds setted.

5. Conclusion and future work

The DALER robot has been simulated on Webots and the cost of transport of walking has been optimised for different ground using a genetic algorithm. By choosing a configuration with a fixed wingspan corresponding to half-opened wings, all other parameters are fixed over the speed of the motors of wingers, the phase shift between these motors and the ratio r between the speed of the motors of wingers and of shoulders.

The first observation made is that the robot is more efficient on surfaces with a high friction coefficient between the ground and the wings. Secondly, two sets of parameters are optimal depending on the form of the terrain. On flat grounds, the speed of motors of wingers is near to 3 rad/s, the phase shift between them is approximately equal to π and the ratio r is equal to 2 or 3. The shoulders and elbows move then symmetrically. For slopes up and rough terrains, the speed of wingers is smaller, about 2 rad/s, the phase is equal to $\pi/2$ or $3\pi/2$ and the ratio r is equal to 1. This gait is therefore non-symmetric and allows the robot to compensate the slipperiness of the terrain by using one winger after the other to “grip” the ground and “climb” the slope.

Future work consists in implementing and testing these gaits on the real robot. It could be interesting to use a hybrid machine learning technic on the robot by taking the parameters found by the algorithm and making again the evolution on real hardware. The parameters would then be more adapted to the robot, which has some small differences with the simulation.

References

- [1] L. Daler, J. Lecoecur, P. B. Hählen, and D. Floreano, “A Flying Robot with Adaptive Morphology for Multi-Modal Locomotion,” in IROS 13, no. EPFL-CONF-187750, 2013.
- [2] L. Daler, S. Mintchev, C. Stefanini, and D. Floreano, “A Bioinspired Multi-Modal Flying and Walking Robot”, Bioinspiration & Biomimetics submission, EPFL-LIS, 2014.
- [3] I. Zeaf, “Ground controller for a flying and walking robot with adaptive morphology”, Semester project in EPFL-LIS, 2014.
- [4] D. E. Goldberg, “Genetic Algorithms in Search, Optimization, and Machine-Learning”, pp 1-23, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, 1989.
- [5] W. Press, S. Teukolsky, W. Vetterling, B. Flannery, “Numerical Recipes in C”, Ch.7, Cambridge University Press, 1992.