Implementing adhesion for NeuroMechFly – Neuromechanical computational model of Drosophila Melanogaster

Laetitia Schwitter
supervised by:
Victor Alfred Stimpfling
Prof. Pavan Ramdya

Bachelor Project
Spring 2023

Contents

1 Abstract **3**

2 Introduction **3**

2.1 The importance of understanding Drosophila Melanogaster 3

2.2 The importance of simulating Drosophila Melanogaster 3

2.3 Project Aim 4

3 State of the art: Adhesion in Insects **5**

3.1 Adhesion in D. Melanogaster: 5

3.2 Gaits for locomotion : 5

4 State of the art: Simulations in a Gym Environment **7**

4.1 NeuroMechFly: 7

4.2 MuJoCo 7

4.3 FlyGym 8

5 Methods: Implementing adhesion for the NeuroMechFly model **9**

5.1 Creating a simple adhesion model : Block 9

5.2 Implementing adhesion using NMF 10

5.2.1 Defining units for NeuroMechFly in MuJoCo 10

5.3 Static: Adhesion On/ OFF on all Legs, single Leg 10

5.3.1 Grid Search to find max gain: Single Leg 10

5.3.2 Behavior of Contact forces with adhesion 11

5.4 Dynamic: Walking 11

5.4.1 Generating custom gaits for flat surface walking 12

5.4.2 Walking with adhesion: OFFline calculation of adhesion 12

5.4.3 Walking with adhesion: INline calculation of adhesion 14

6 Results **15**

6.1 Creating a simple adhesion model : Block 15

6.2 Static Fly : Adhesion using NeuroMechFly 15

6.2.1 Grid Search to find max gain: Single Leg 15

6.2.2 Behavior of Contact forces with adhesion 16

6.3 Dynamic: Walking with adhesion 18

6.3.1 Results from smoothing the contact forces: tarsus5 vs sum of all tarsi 18

6.3.2 Adhesion Vector calculated before the simulation: Adhesion rule: $>$ 19

6.3.3 Adhesion Vector calculated before the simulation: Adhesion rule: $\geq$ 19

6.3.4 InLine adhesion (Iterative Method) and Adhesion rule: $\geq$ 24

7 Discussion **26**

7.1 Static: Adhesion On/ OFF on all Legs, single Leg and tripod 26

7.1.1 Grid Search to find max gain: Single Leg 26

7.1.2 Behavior of Contact forces with adhesion 26

7.2 Dynamic: Walking with adhesion 27

7.2.1 Prerun adhesion vector calculation 27

7.2.2 Inline adhesion vector calculation 28

7.2.3 Further adhesion rules for future experiments 28

8 Conclusion **28**

9 Gantt Chart **29**

10 Appendix **31**

1 Abstract

Adhesion is important for walking smooth and rough surfaces. NeuroMechFly, a neuromechanical computational model of the Drosophila Melanogaster, did not include any possibility of adhesion. In this project the linear relationship between the contact forces and the adhesion strength was confirmed in the aim of developing adhesion rules that work well for normal and inverted gravity walking.

2 Introduction

Locomotion is an important ability that animals use to navigate their environment in the search of food, a mate and to escape dangers.

It can take many different forms among the species, depending on the number of legs. This gives rise to a computational challenge : how can the body of an animal be modeled efficiently for realistic locomotion? The best option we have to this day is to model legs as a combination of joints. For a multi-legged species, each joint then has to be coordinated with the rest of the body to allow coordinated movement. This in turn has to be adapted based on the wanted velocity or on the characteristics of the terrain.

2.1 The importance of understanding Drosophila Melanogaster

Drosophila Melanogaster are hexapods that are very commonly used to study complex animal behavior [1].

Their success as model organisms for developmental biology is sourced in the genetic tools that can be applied on them, their fast reproduction rate, and their low production cost [8].

Similar arguments apply for their use as a model organism for brain diseases where many fly genes act homologous to human genes. Drosophila Melanogaster also possess a highly complex neural network that allows for locomotion and complex behaviors such as walking and grooming.

2.2 The importance of simulating Drosophila Melanogaster

The D. Melanogaster is very small (on average: 3 mm in length and 2 mm in width [10] and leg mass $m = 1.42 \times 10^{-}7$ kg, [3]). This makes the study of their behavior challenging, as the measure of angles, forces and torques requires extremely precise devices. Obtaining detailed measurements are necessary to understand how exactly the fly manages to stand, walk, climb, etc...

Changing a single parameter such as friction, adhesion, joint lengths or the terrain experimentally is generally linked with high expenses, having to create mutations, and obtaining high spatial and temporal resolution footage for very specific tasks that then need to be analyzed.

With an accurate model of the D. Melanogaster, changing single parameters for a simulation becomes a lot easier compared to changing the same parameters experimentally. The relevant values found can then be validated experimentally. Therefore, simulating the fly can help understand fly behavior and how its behavior is influenced by the nature of the external environment and its own neuromechanical system. This was done by studying movement control of the animal [1].

2.3 Project Aim

The lab aims to study the movement control of D. Melanogaster [1]. It was found in [3] that blocking the adhesive properties of the fly experimentally results in the fly adopting an atypical bipod-like coordination instead of a tripod gait. Adding adhesion could consequently allow it to retrieve similar results and help in optimizing gaits for speed and stability ([1]). The current model does not include any adhesive properties on the Tarsi. This is a limitation if simulations on more complex terrain such as rough or vertical surfaces need to be performed. This is why this project aims at laying the groundwork for flat surfaced fly locomotion using adhesion. For this task, adhesion was added to the tarsus 5 of each leg, the relationship between adhesion and the contact forces during the simulation were studies, and different rules for adhesion were tested.

3 State of the art: Adhesion in Insects

3.1 Adhesion in D. Melanogaster:

Adhesion in Drosophilia Melanogaster is still an active field of research. In 2015, [9] stated that the climbing of vertical surfaces in insects is possible mainly due to their claws and the smooth or hairy adhesive pads on the pretarsus (Figure 1). The adhesive pads take advantage of capillary forces and Van der Waals forces for wet adhesion, and enable the insects to walk on vertical smooth surfaces. While the claw is not an adhesive structure per say, it is involved in the climbing of rough grounds.

Also, although it is known that for flies, the main adhesive structure is the pulvillus, which is covered by hair-like structures called setae that can be found on each leg Figure 1. Only little data has been collected on the precise workings of the adhesive device in Drosophila Melanogaster due to their small size. However, during their study, [9] found that Drosophila secrete a liquid at the setae contact zones to produce capillary forces for adhesion. Additionally, by comparing a mutant ($Su(z)2$) that contained malformed and/or shorter setae compared to the wild type, it was found that the mutation reduced the ability of the fly to walk on vertical smooth surfaces, but that their ability to walk on horizontal grounds remained unaffected.

When increasing the roughness of the vertical surface, the claw was able to counteract much of the lost adhesion capacity, so that the fly was still able to climb. However, when [3] blocked the pretarsus with an ultraviolet polymer both the claw and pulvillus were not accessible for the fly. This caused the predominant fly gait to become an atypical bipod gate instead of the normal tripod gait. Similarly, they found that when covering the more proximal tarsal segments, the gait displays high TCS (Tripod Coordination Strength). [3] also measured that adhesion in a single leg allows 9 flies out of 10 to hang from the wall for more than 1 second.

3.2 Gaits for locomotion :

Drosophila, like other hexapods, show a variety of coordinated movement types, with no proven preferred configuration when subjected to a varying environment ([2]). However, the changes in gait are done on a continuous spectrum.

[2] characterized these changes into symmetric and asymmetric variability. The former is due to changes in walking speeds, and mostly consists of out of phase symmetric changes for both sides of the body, while the latter is due to multiple smaller changes such as asymmetric joint angles that allow the insect to turn.

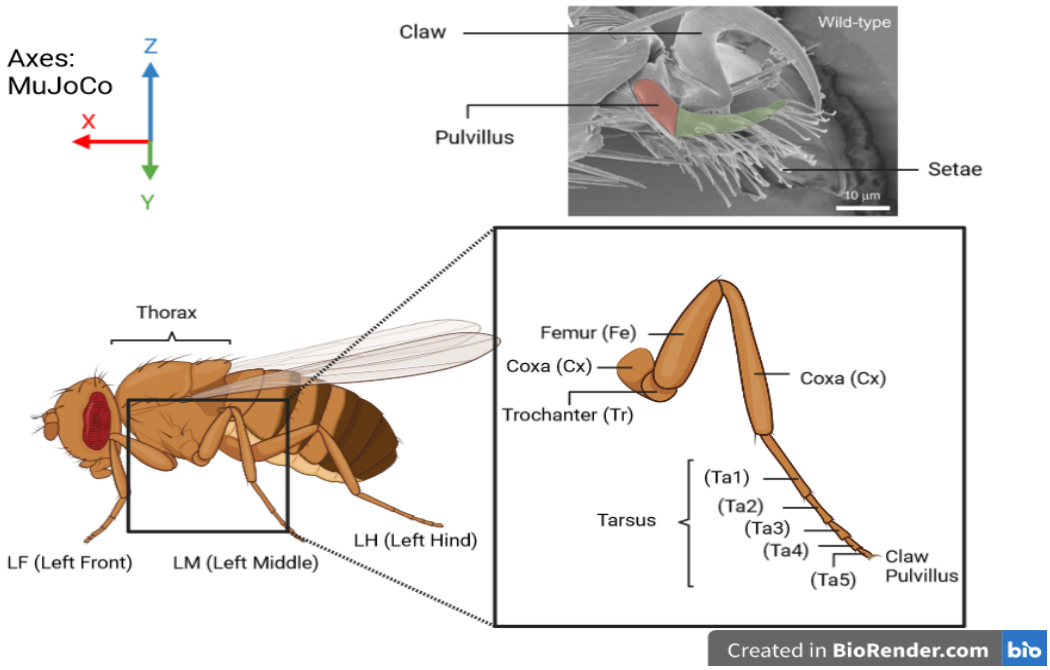

Figure 1: Shows a labeled model of D. Melanogaster in the cartesian coordinates used in MuJoCo. The image of the adhesive structures of D. Melanogaster were annotated from ([9])

Walking speeds and stance duration affects the gait. Previous studies have shown that an increased stance duration and a higher number of legs in stance correlates with a slowdown in walking speed, whereas swinging duration had no influence [2].

The Drosophila Melanogaster is known to transition from a slow wave gait to a tetrapod gait and then to a tripod gait at high walking speed [2], [3]. Unlike in vertebrates, the transitions are done in a more continuous manner and no matter the speed, the number of limbs in contact with the floor rarely go below three [3]. The exact reason as to why the tripod gait is so popular is unknown, but it is assumed that it allows the fly to cover large distances on complex terrain, even with vertical components.

Experimentally [3] also found that blocking the adhesive properties of the fly results in the fly adopting a atypical bipod-like coordination instead of a tripod gait.

For the tripod gait, the front and hind legs of one side are simultaneously in swing with the middle leg from the other side, always keeping the 3 other legs in stance. Locomotion is generated by the alternation of the leg groups that are in swing [2].

4 State of the art: Simulations in a Gym Environment

4.1 NeuroMechFly:

NeuroMechFly is a neuromechanical model that simulates the adult Drosophila melanogaster (fruit fly). The model consists of a physics-based simulation environment, a biomechanical exoskeleton, muscle models, and neural network controllers [1], [6]. At the beginning of the project, NeuroMechFly was mainly run on the physics engine Pybullet and partially on IsaacGym.

The model achieved several improvements with respect to previous fly models. It now takes into account mass distributions, all 7 necessary degrees of freedom (DoF) per leg for realistic leg movement, and other physical constraints [1]. The 3D flies used to generate the biomechanical model were scaled to relative ThC positions and average body part lengths [1]. This way, joint angles could be defined in roll-pitch-yaw radian angles instead of in 3D Cartesian coordinates. To avoid low-magnitude numerical errors, the model was scaled dynamically along with the physical parameters (e.g., gravity). The model was developed to simulate behaviors such as walking and grooming. For each step of the simulation, joint angles are inputted into the biomechanical model. These joint angles were obtained through 3D pose estimation of labeled frames of a tethered fly walking on a ball and not on a flat surface. Joint compliance in the model was achieved by adapting kp values, stiffness and damping parameters, which control the flexibility and the movement-resistance of the joints.

The current NeuroMechFly Biomechanical model does not model the pulvillus nor the claw.

4.2 MuJoCo

MuJoCo is a physics engine that enables physics driven simulations [4]. It provides various actuators that allow grouping certain elements with similar functions for facilitated access during runtime. Any specific actuator can be accessed through its name. The adhesion actuator specifically applies an adhesion force linearly in the direction normal to the contact force onto all geometries (called `geoms`) that are associated with a specific body. The total adhesion force of the actuator is obtained by multiplying the gain of the actuator with the control value.

To facilitate the use of MuJoCo for reinforcement learning tasks, DeepMind has developed `dm_control`, which is a MuJoCo wrapper containing Python libraries [5]. The adhesion actuator in MuJoCo can either act only when the body is in contact with another body or as an attractive force, similar to an electromagnetic attractive force [4]. For NeuroMechFly, adhesion should not have this property and only be active when the tarsi are in contact with the ground.

4.3 FlyGym

During the course of the project, the lab has worked on developing FlyGym for MuJoCo and other Gym environments. It was implemented with the goal of using NeuroMechFly for reinforcement tasks [7]. The model can be instantiated and visualized easily. The `step()` function updates the simulation and yields observations of the state of the simulation. The observations contain information about the contact forces, cartesian fly positions, velocities and end_effectors positions. The end effectors refer to the tarsus 5.

5 Methods: Implementing adhesion for the NeuroMechFly model

5.1 Creating a simple adhesion model : Block

As a proof of concept, the fly model was abstracted using a simple block. This was done to get accustomed with MuJoCo and `dm_control`. The main aim was to get familiar with the structure of XML files, and to learn how to access positional arguments or gravity, and to implement adhesion.

The adhesion was added on the level of the biophysical model of the block (i.e. in the XML) and during the initialization of the model.
NB: The actuator must be added before the physics environment is created

Here is a code example where the initial gain of the adhesion actuator is set to 1000, however the final strength will be equal to $adhesion_value \times gain$:

```
#initializing an adhesion actuator for a specific body
adhesion_actuator = self.model.actuator.add('adhesion',
                                            name =f"{body_name}_adhesion",
                                            gain=1000,
                                            body=body_name,
                                            ctrlrange="0 1000000",
                                            forcerange="-inf inf")
#Initialize the rest of the model
....
#create the physics environment
....
#Accessing and changing the control value of the actuator to the wanted adhesion_value.
self.physics.bind(adhesion_actuator).ctrl = adhesion_value
```

Gravity can be accessed and modified like this:

```
nmf.physics.model.opt.gravity = (0,0,+9.81)
```

The initial position of the block was on the ground. The gravity was inverted. The adhesion actuator remained on for a while, then it was turned off.

The implementation can be found on: `BrickOnCeiling.ipynb`

5.2 Implementing adhesion using NMF

No data of an untethered fly walking on a flat surface has yet been recorded. The preexisting data comes from the fly walking on a ball. Nevertheless, the implementation of adhesion is concentrated on a fly walking on flat terrain. During the course of this project, the lab has developed FlyGym, a reinforcement tool that allows users to run NeuroMechFly in a simple and concise manner. In order to maintain its reinforcement aspect, an action dictionary is inputted at each step. The adhesion actuators were added to a specific body with a chosen gain during the initialization of the model. The exact adhesion values are computed externally and later added to the action function.
An adhesion key was added to the action dictionary so that at each timestep the `action["adhesion"]` now inputs as parameter an array of size 6 whose values correspond to `['LFTarsus5', 'LMTarsus5', 'LHTarsus5', 'RFTarsus5', 'RMTarsus5','RHTarsus5']` respectively.

The claw and the pulvillus are the main structures for the flies' adhesion-like properties. As they are not yet part of the NeuroMechFly model, the adhesion was added to the tarsus 5 of each leg as the pretarsus is most relevant for adhesion [3].

It must be noted that for each experiment the fly was left to stabilize for 1000 timesteps.

5.2.1 Defining units for NeuroMechFly in MuJoCo

In MuJoCo, the NeuroMechFly model maintains the mass and time unscaled; however, the fly's size was scaled by a factor of 10^6 in all dimensions. This means that 1m in MuJoCo corresponds to $1\mu m$ in reality. Similarly, for the forces, $1N$ would correspond to $1\mu N$. However, the gravity that appears to work for NeuroMechFly in FlyGym is not consistent with this scaling as it is only scaled by a factor of 10^5, i.e. $(0, 0, -9.85 \times 10^5)$. To avoid any inconsistencies due to units, the report will use $[force]$ for the unit of force in MuJoCo and $[length]$ for the unit of length / distance.

5.3 Static: Adhesion On/ OFF on all Legs, single Leg

5.3.1 Grid Search to find max gain: Single Leg

The adhesive structures allow the fly to stick to the ceiling with a single leg [3]. To find the minimal force necessary for it to hold with a single leg in the simulation, a gridsearch was performed on a fly with an asymmetric initial pose (Figure 2). Ideally, during the simulation the input adhesion control values are between 0 and 1. For this, it is necessary to find the critical gain at which a single leg allows the fly to remain attached to the ground when gravity is inverted.

The implementation can be found on the Jupyter Notebook `mujoco_singleleg_grid.ipynb`

Figure 2: Initial asymmetric tripod stance (`tripod_engineered.pkl`)

NB: During the implementation of the gridsearch the joint angles of the initial pose were inputted back into the simulation at each time step.

5.3.2 Behavior of Contact forces with adhesion

Adhesion force is supposedly added linearly in the direction normal to the contact force ([4]). To test the reactivity of adding adhesion on the simulation, as well as to derive the relationship between contact forces and adhesion, the latter was added to all legs for increasingly long time intervals. The adhesion actuators on the tarsi5 are initialized with a gain of 4000, which roughly corresponds to the maximum value found in subsubsection 5.3.1. The joint angle data set used was `tripod_engineered.pkl`.

The Jupyter notebook `mujoco_contact_forces.ipynb` tests this in two steps:
When downward gravity is applied, we first test whether adhesion increases the contact forces then we take interest in how they are related. Indeed, it seems natural to think that the contact forces increase by the amount of added adhesion.

Furthermore, contact forces are expected to be quite noisy, therefore a median filter is applied to denoise and smooth the forces. A window of 201 timesteps is used for this. Having a smoothed force vector can later be used to calculate the time derivative necessary for the adhesion rules defined in (subsubsection 5.4.2).

5.4 Dynamic: Walking

Since the ultimate goal is to have flies walking with adhesion, it becomes necessary to identify appropriate rules for adhesion. These rules were tested on both normal gravity and inverted gravity by tracking the trajectory of the end-effectors, the distance travelled by the thorax and the contact

forces.

5.4.1 Generating custom gaits for flat surface walking

Since the joint angles collected from the fly walking on a ball are not necessarily suitable for the fly to walk on a flat surface, the `loop_single_step.ipynb` notebook attempts to create a more dynamically favorable stance for the fly.
Several methods were developed in the `loop_single_step.ipynb` notebook to generate a tripod gait, and in the `loop_single_step_bipod.ipynb` notebook to generate a bipod gait.

`single_steps.pkl` contains the data for a single step cycle and was used to generate the new joint angle data sets.

Since the tripod gait has two powerstrokes, the leg pairs (left, hind leg from one side, middle leg from the other side) were shifted by a 180° phase (i.e. half a cycle) with respect to the other leg pair. Similarly, the bipod gait has three powerstrokes. One can observe by visual inspection of live videos of a fly that, in real life, the fly appeared to lift the hind legs more. To generate such data and improve the fly's stability, specific joint angles were amplified by a constant factor:

$$x(t) = x_{\text{mean}} + \gamma \cdot \big(x(t) - x_{\text{mean}}\big)$$

using the function `amplify_angle_mvmt(joint_name, data, factor)`.

Here, the mean angle is kept constant. Since it is expected that the middle legs will not have the same use on a ball compared to a flat surface, the angles were corrected manually using the function `autocorrect_angles(data)`

NB: Adding 5° to the pitch component of a joint is symmetric for both left and right legs. However, adding 5° to either the yaw or roll component of a left joint, is equivalent to subtracting 5° to the equivalent same joint on the right side. The obtained angles were then used for the following experiment.

5.4.2 Walking with adhesion: OFFline calculation of adhesion

The goal was to find appropriate conditions for the fly to walk a few steps with adhesion on. The NeuroMechFly model was initialized with an initial gain of 8000. The control value of adhesion 0 or 1 was then multiplied by a factor `adhesion_gain_multiplier` to change the magnitude of the adhesion force.

First, the simulation was run without adhesion. The contact forces, the distance and the end effector positions were studied. The contact forces were filtered with a median filter of window size 501 timesteps (again, several different window sizes were tested). The effect of both summing the contact forces of all tarsi of the same leg and the effect of only looking at the tarsi5 were implemented. The derivative of the forces was calculated on the filtered contact forces.

Two different adhesion rules were tested:

1. the first one was to apply adhesion only when derivative was positive or equal to zero: $\frac{dF_{\text{contact}}}{dt} \geq 0$
2. the second one applied adhesion only when the derivative was strictly positive: $\frac{dF_{\text{contact}}}{dt} > 0$

This makes sense as a negative derivative means that the force diminishes: this should correspond to the fly lifting its leg.
Also, adhesion forces are contact forces: they should only influence the fly when the leg is in contact with the floor.

According to these rules, the adhesion vector was calculated and later inputted for both the normal and inverted gravity walking.

To have a direct comparison between the different adhesion strengths without having to reinitialize a model each time, the adhesion control value was multiplied with different `adhesion_gain_multipliers`. This allowed for two variations of calculating the adhesion vector using the contact forces of a previous simulation:

1. **Simple method:** The adhesion vector was calculated based on the median filtered contact forces of the fly walking **without** adhesion and with normal gravity.
2. **Iterative method:** The adhesion vector was calculated anew for each `adhesion_gain_multiplier` based on the median filtered contact forces of the previous fly walking. When simulating the fly with normal gravity, the contact forces of the simulation with a lower gain are used (i.e. with a smaller `adhesion_gain_multiplier`). For the simulations with inverted gravity, the second simulation is the one with the highest gain and uses the contact forces of the first simulation (without adhesion).

5.4.3 Walking with adhesion: INline calculation of adhesion

The NeuroMechFly model is initialized the same as in the previous section. However, the adhesion vector is calculated dynamically within the simulation (inline) by computing the derivative of the contact forces using the median of the previous steps. The `window_size` for this part of the experiment was 501. Again, different `adhesion_gain_multipliers` were used to compare different adhesion forces.

6 Results

6.1 Creating a simple adhesion model : Block

The video where the adhesion of the block that stands on the ground with inverted gravity was turned off, can be found on the git repository of this project.

The actuator was both added externally and directly in the XML file.

6.2 Static Fly : Adhesion using NeuroMechFly

6.2.1 Grid Search to find max gain: Single Leg

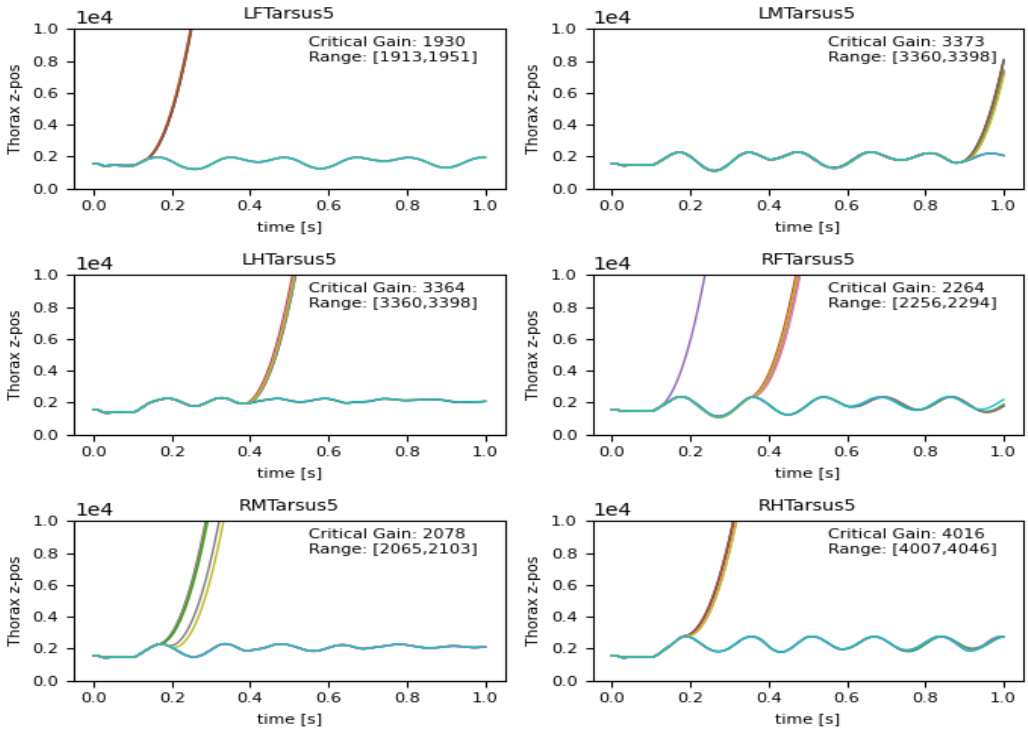

Figure 3: Shows the Single leg adhesion gain grid search with inverted gravity at 0.1s, the joints are maintained so that the fly remains in the asymmetric tripod pose. When the thorax position is higher than 2000[*length*] the fly has escaped the equilibrium position and flown away.

The grid search managed to find the critical gain for each leg at which the fly did not fly away once gravity was inverted Figure 3. The maximum critical gain was on the right hind tarsus 5 (adhesion gain: 4016). The critical gains are similar but not perfectly symmetric between both sides of the body. This is because the leg angles are shifted by a half-cycle (See initial stance: Figure 2). Some tarsi5 are located further away from the center of mass. The fly therefore requires higher adhesion to counteract the imbalance induced by the joint angles.

The video for the simulation additionally shows that the fly maintains the fly joints as imposed by

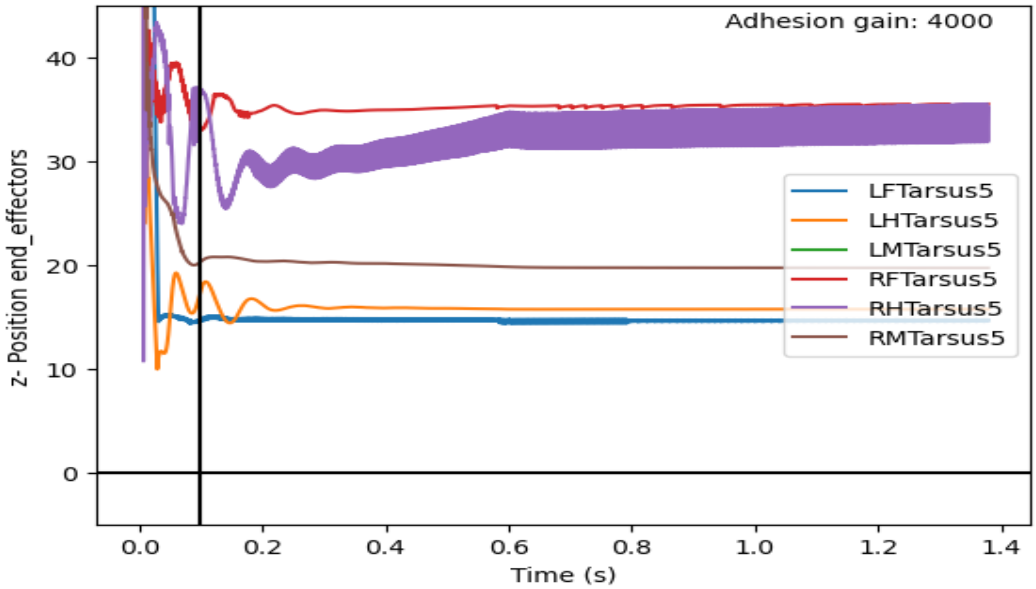

Figure 4: This figure shows the static fly at inverted gravity with adhesion (adhesion strength: 4000) applied only to the left-front tarsus 5, left-hind tarsus 5, and right-middle tarsus 5 (tripod).

the given data set. The body of the fly rotates around the active tarsus 5 to align the center of gravity with the two forces acting on it.

To validate whether the maximum critical gain (adhesion gain: 4000) makes the legs traverse the floor, adhesion was applied only on the left-front tarsus 5, left-hind tarsus 5, right-middle tarsus 5 (Figure 4). Their end effector positions stabilize at a height within the range of $[10, 15][length]$ This is due to the imposed range in FlyGym for contact detection between the floor and the leg.

6.2.2 Behavior of Contact forces with adhesion

When adding adhesion, the contact forces at first increase by more than the added adhesion strength. The contact forces are also lower immediately after removing adhesion. The fly manages to stabilize this overshoot in both the case of the Tarsi 5 (Figure 5) and when summing of all tarsal contact forces (Figure 6) after less than 1000 timesteps (i.e. $0.1s$). For shorter intervals, the contact forces do not stabilize entirely. The contact forces of the tarsus 5 (Figure 5) do not increase exactly by 4000. This effect is less visible when looking at the sum of the contact forces of all tarsi (Figure 6).

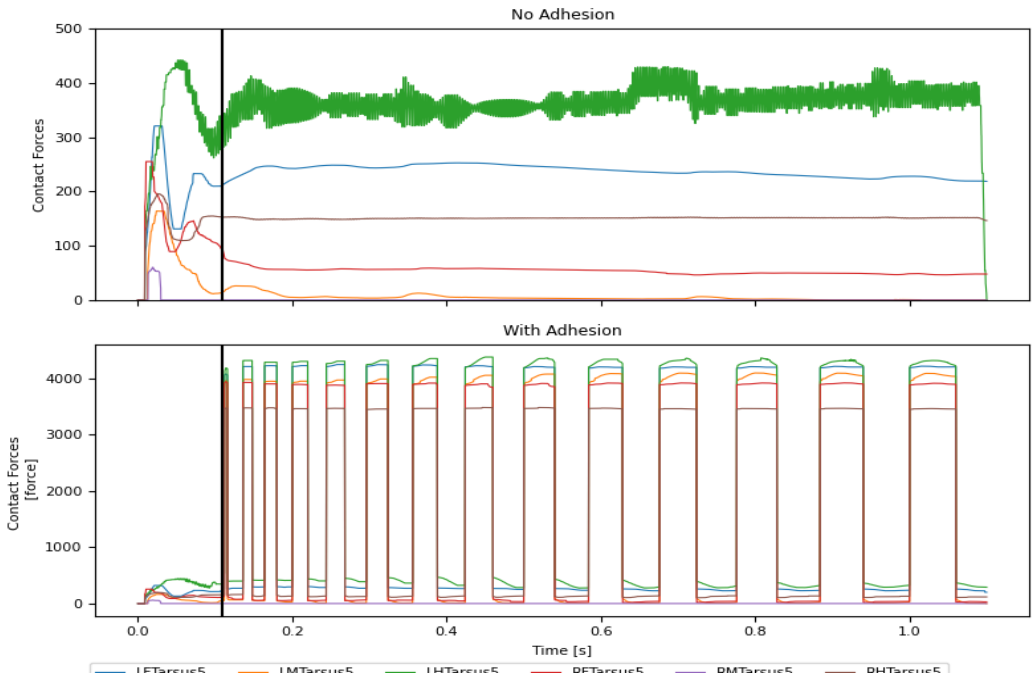

Figure 5: Comparison of static fly adapting to the addition of adhesion. The adhesion gain is 4000. The contact forces are those of only the Tarsi5 and are filtered with a window size of 201. The unfiltered data can be found in the (Appendix: Figure 17).

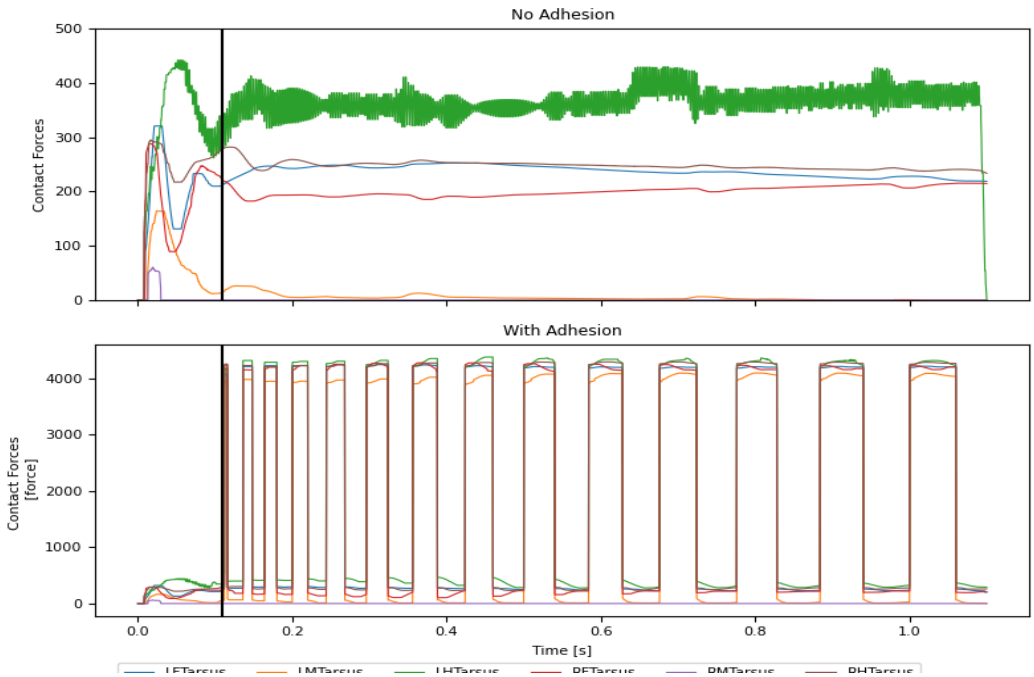

Figure 6: Comparison of static fly adapting to the addition of adhesion. The adhesion gain is 4000. The contact forces are the sum of all Tarsi contact forces filtered with a window size of 201. The unfiltered data can be found in the (Appendix: Figure 18).

6.3 Dynamic: Walking with adhesion

6.3.1 Results from smoothing the contact forces: tarsus5 vs sum of all tarsi

The contact forces from the fly walking without adhesion at normal gravity are very noisy. The contact forces were smoothed. For the same window size, the smoothed contact forces obtained from all tarsi (Figure 8) have a much higher peak than those obtained from only the tarsus 5 (Figure 7).

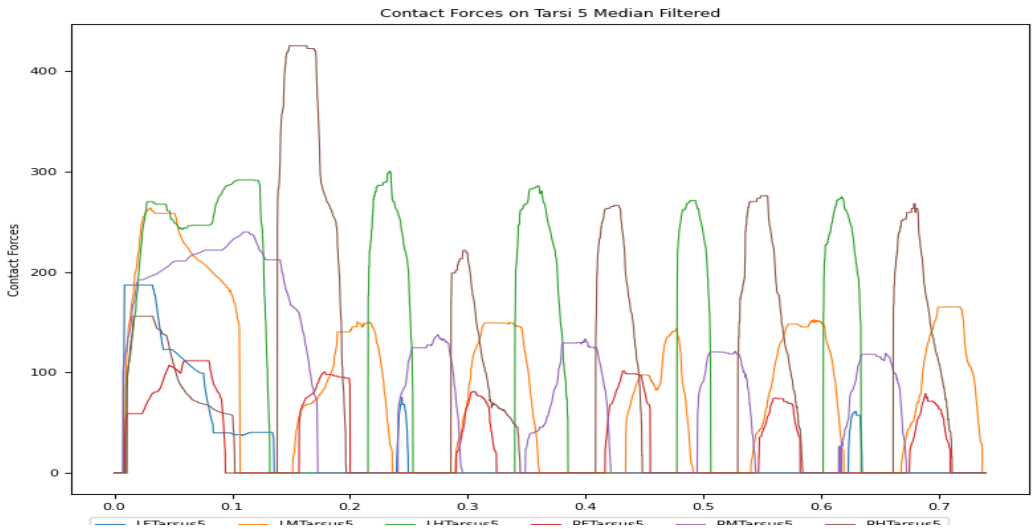

Figure 7: The contact forces are those of only the Tarsi5 when the fly is walking without adhesion at normal gravity. The data is filtered with a windowsize of 501. The unfiltered data can be found in the (Appendix: Figure 19).

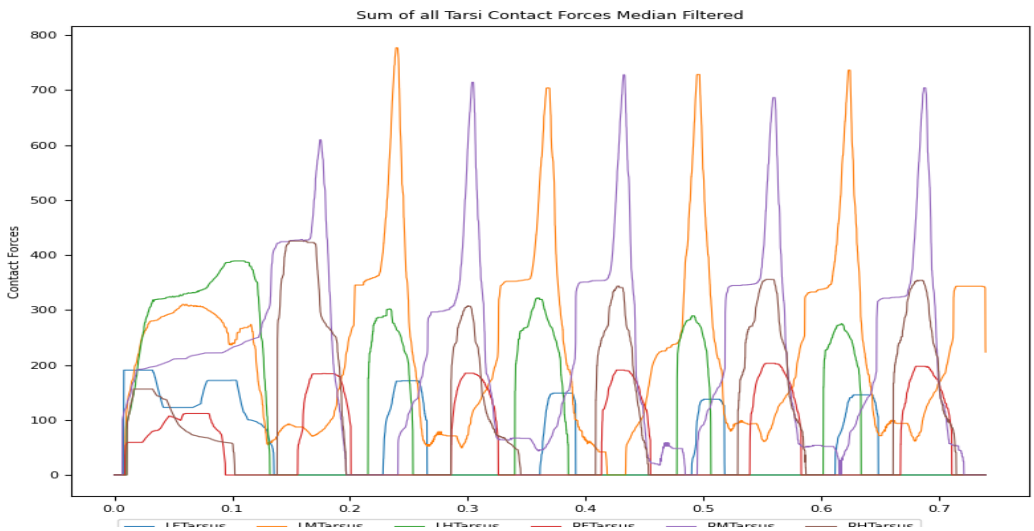

Figure 8: The contact forces are the sum of the contact forces of all tarsi when the fly is walking without adhesion at normal gravity. The data is filtered with a windowsize of 501. The unfiltered data can be found in the (Appendix: Figure 8).

6.3.2 Adhesion Vector calculated before the simulation: Adhesion rule: $>$

All simulations with normal gravity tested within an adhesion gain range of $[400, 5600]$ manage to walk using the Simple Method (Figure 9), although the fly seems highly unstable at the gains 872 and 1445. 2/8 of the flies with adhesion fall to the back. The Iterative Method shows similar results, however the fly appears to fall only after a few steps for the majority of the tested gains (1/8 of the flies with adhesion fall). Figure 11 shows that until the gain of 3709 the fly manages to lift the legs similar to the case without adhesion, however after a few steps the movement of the Tarsi 5 is disrupted.
The distances traveled are lower but similar to without adhesion using the Simple Method of prerun adhesion (Figure 10). With the iterative method the flies travel even less far. The fly with adhesion gain 400 manages to surpass the distance traveled without adhesion, but falls in the end(Figure 12).

When gravity is inverted, both the Simple and the Iterative Method (defined in subsubsection 5.4.2) do not yield an appropriate adhesion vector for walking and the D. Melanogaster flies away (Appendix: Figure 23, Figure 27).

6.3.3 Adhesion Vector calculated before the simulation: Adhesion rule: $\geq$

When gravity is inverted, both the Simple and the Iterative Method (defined in subsubsection 5.4.2) do not yield an appropriate adhesion vector for walking, and the D. Melanogaster flies away (Appendix: Figure 24, Figure 28).
With this adhesion rule, 5/8 of the total flies fall for both the Simple Method and the Iterative method (Appendix: Figure 22, Figure 26). The flies do not advance (Appendix: Figure 21, Figure 25).

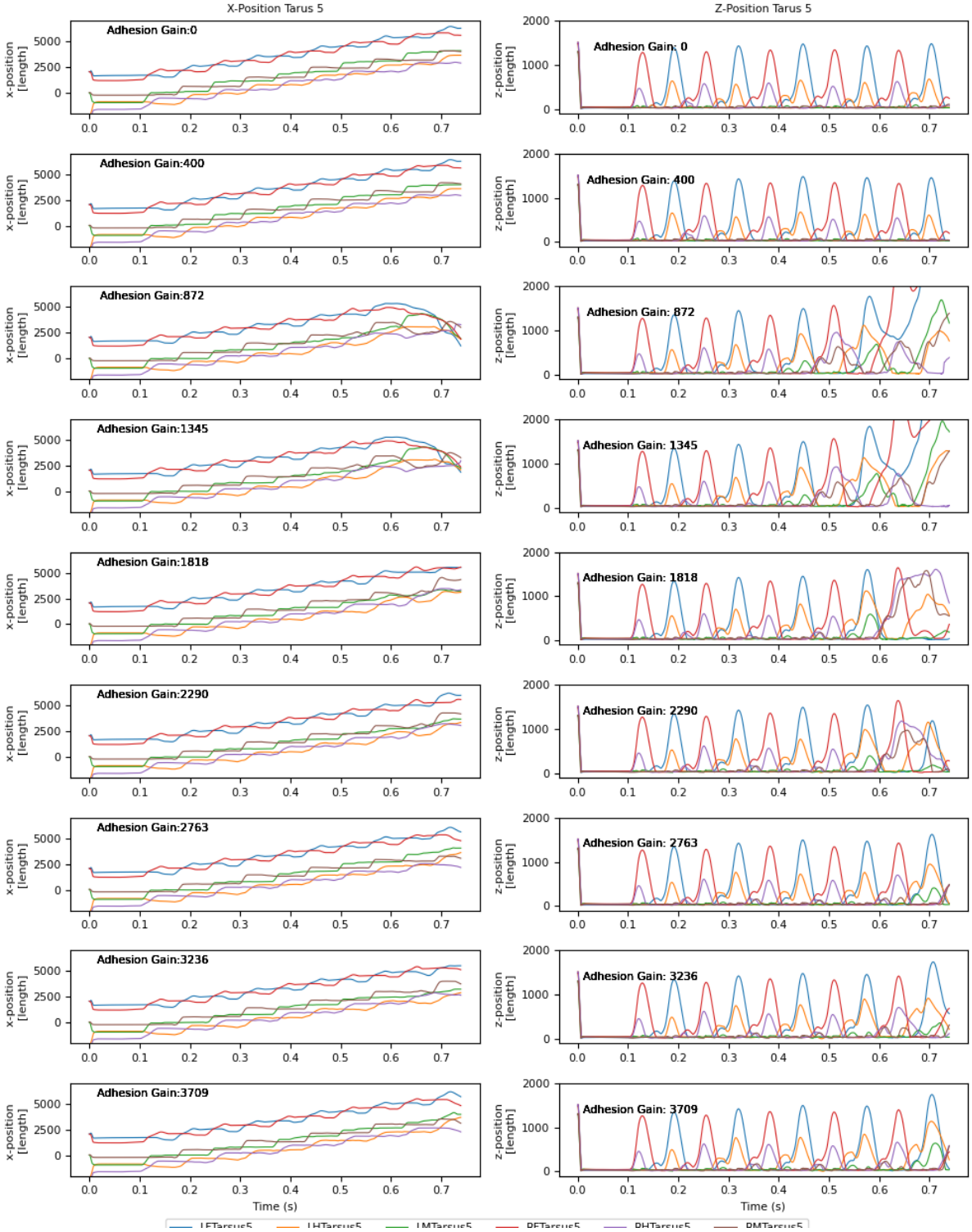

Figure 9: End effector positions travelled after using the Simple Method on different gains for walking with adhesion using the following conditions: Normal Gravity, adhesion rule: " > ".

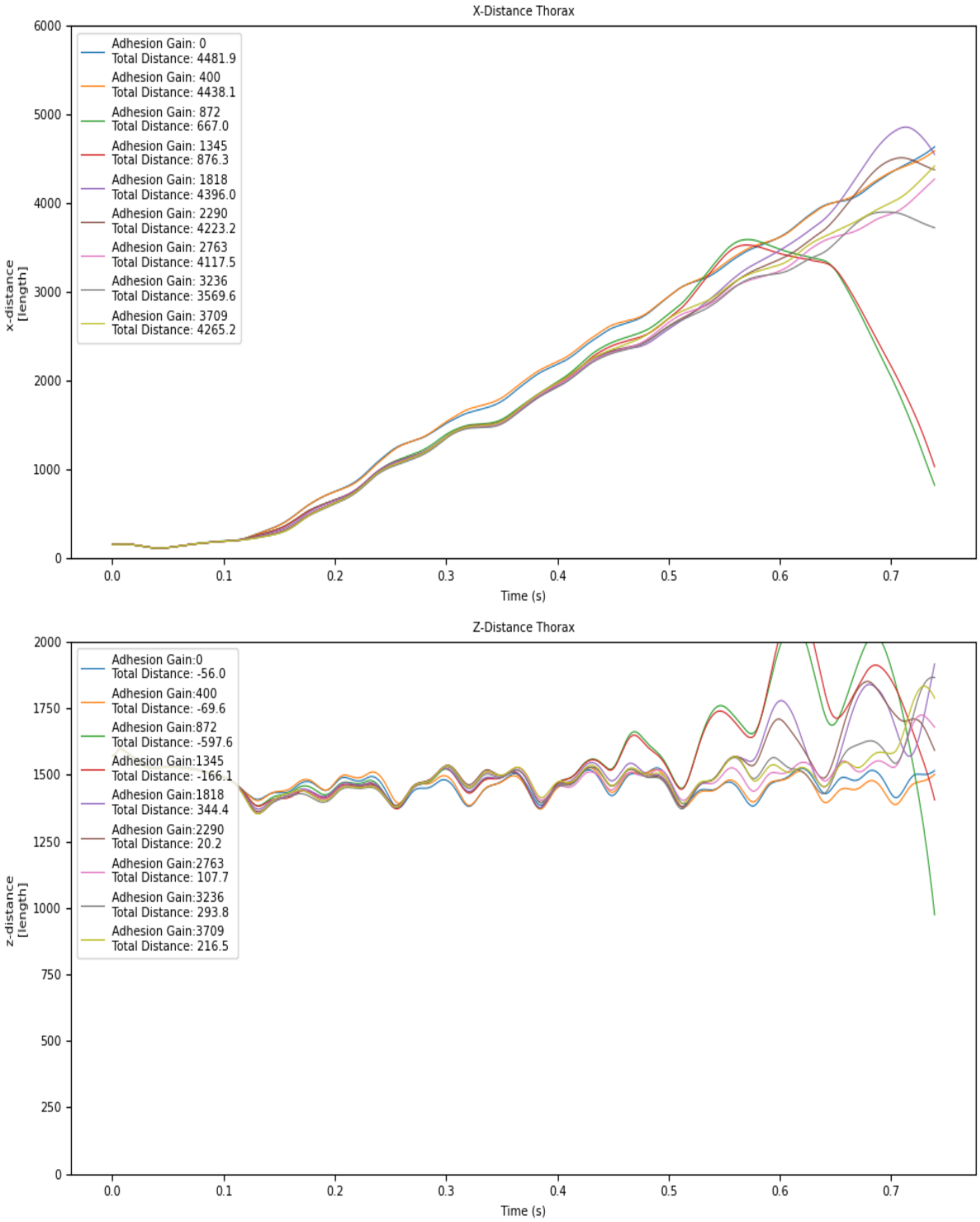

Figure 10: Distances travelled after using the Simple method on different gains for walking with adhesion using the following conditions: Normal Gravity, adhesion rule: " > ".

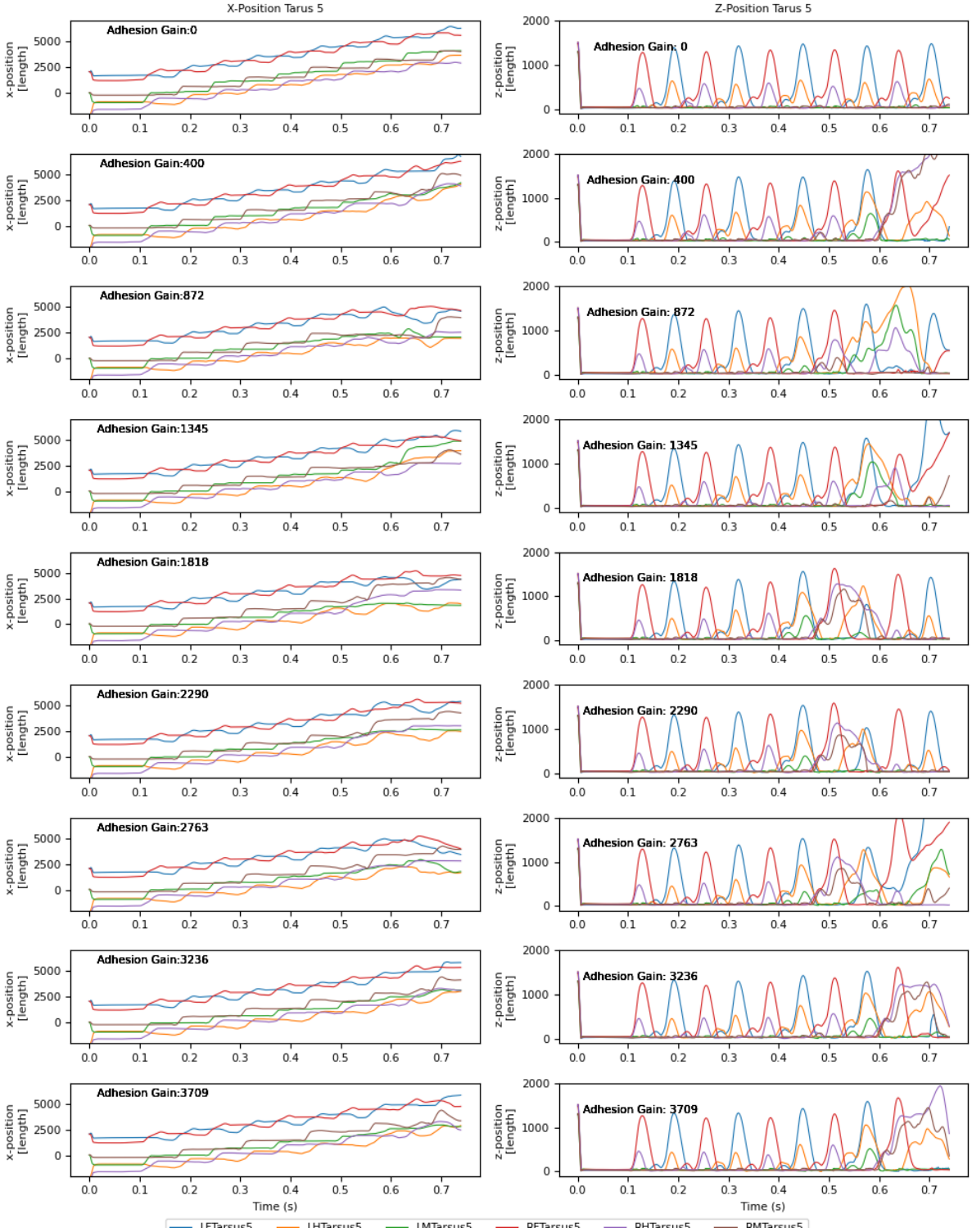

Figure 11: End effector positions travelled after using the Iterative Method on different gains for walking with adhesion using the following conditions: Normal Gravity, adhesion rule: " > ".

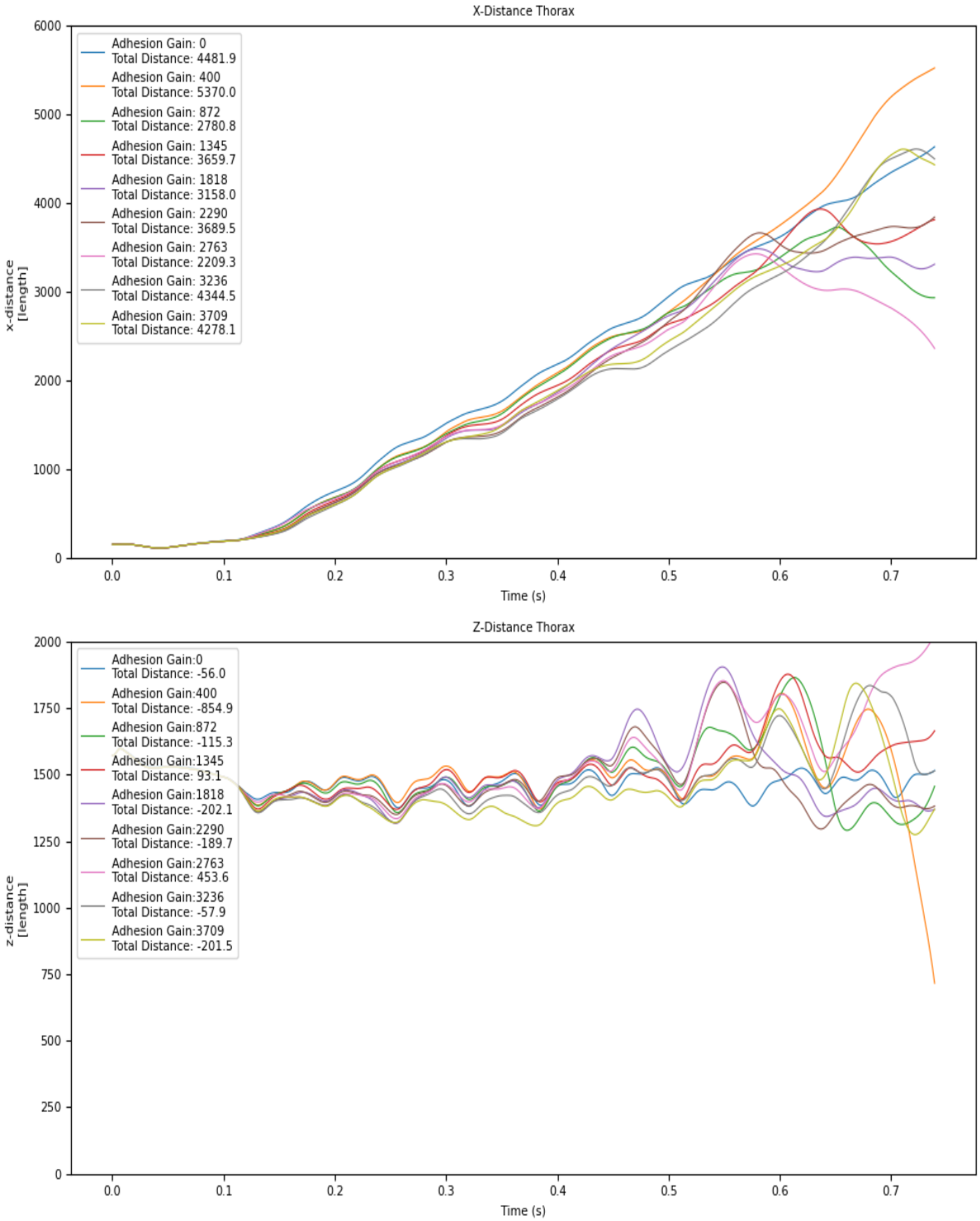

Figure 12: Distances travelled after using the Iterative method on different gains for walking with adhesion using the following conditions: Normal Gravity, adhesion rule: " > ".

6.3.4 InLine adhesion (Iterative Method) and Adhesion rule: $\geq$

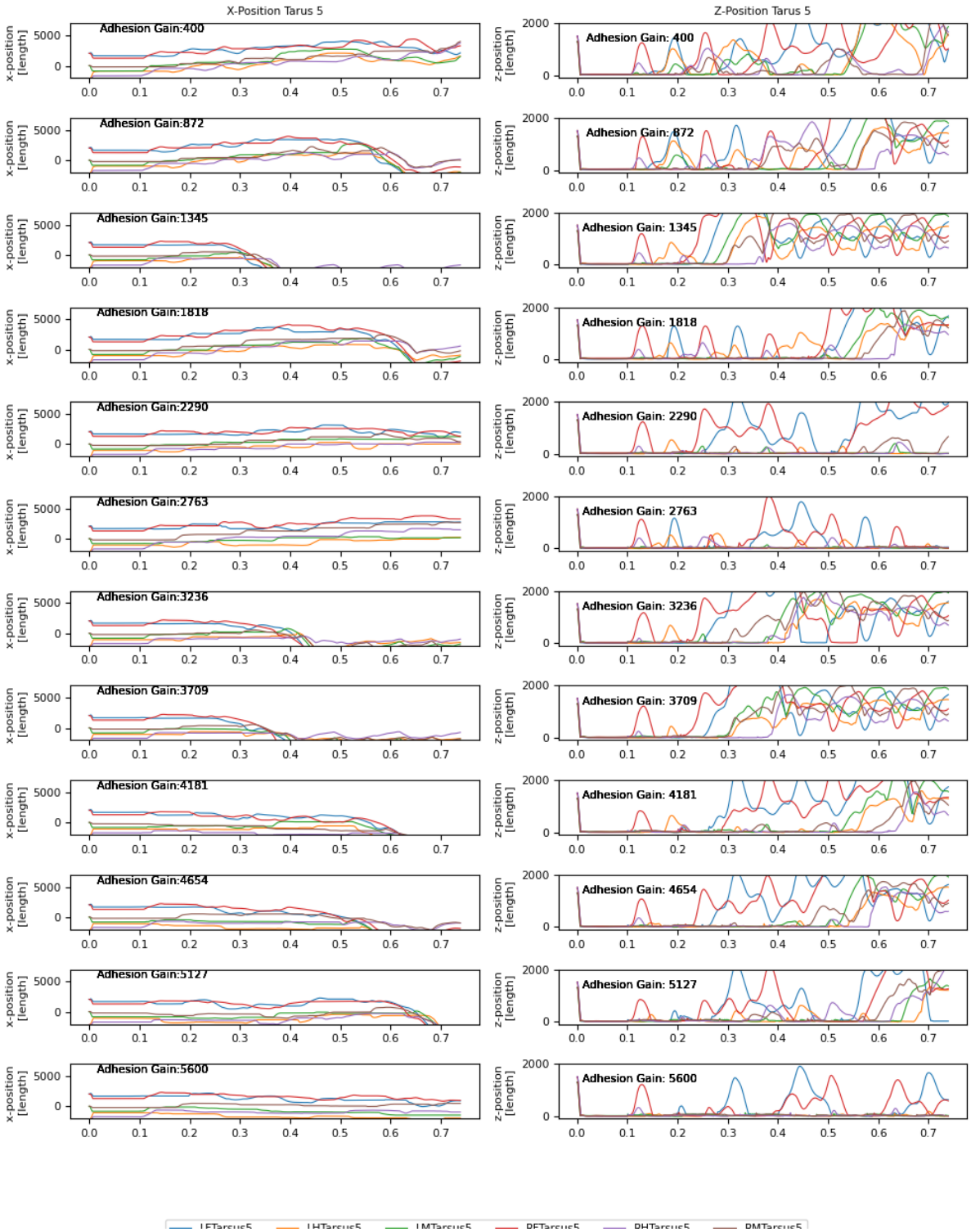

Figure 13: End effector positions travelled after calculating the adhesion vector during the simulation with windowsize 501. The method was applied on different adhesion gains for walking with adhesion in the following conditions: Normal Gravity, adhesion rule: " $>$ ".

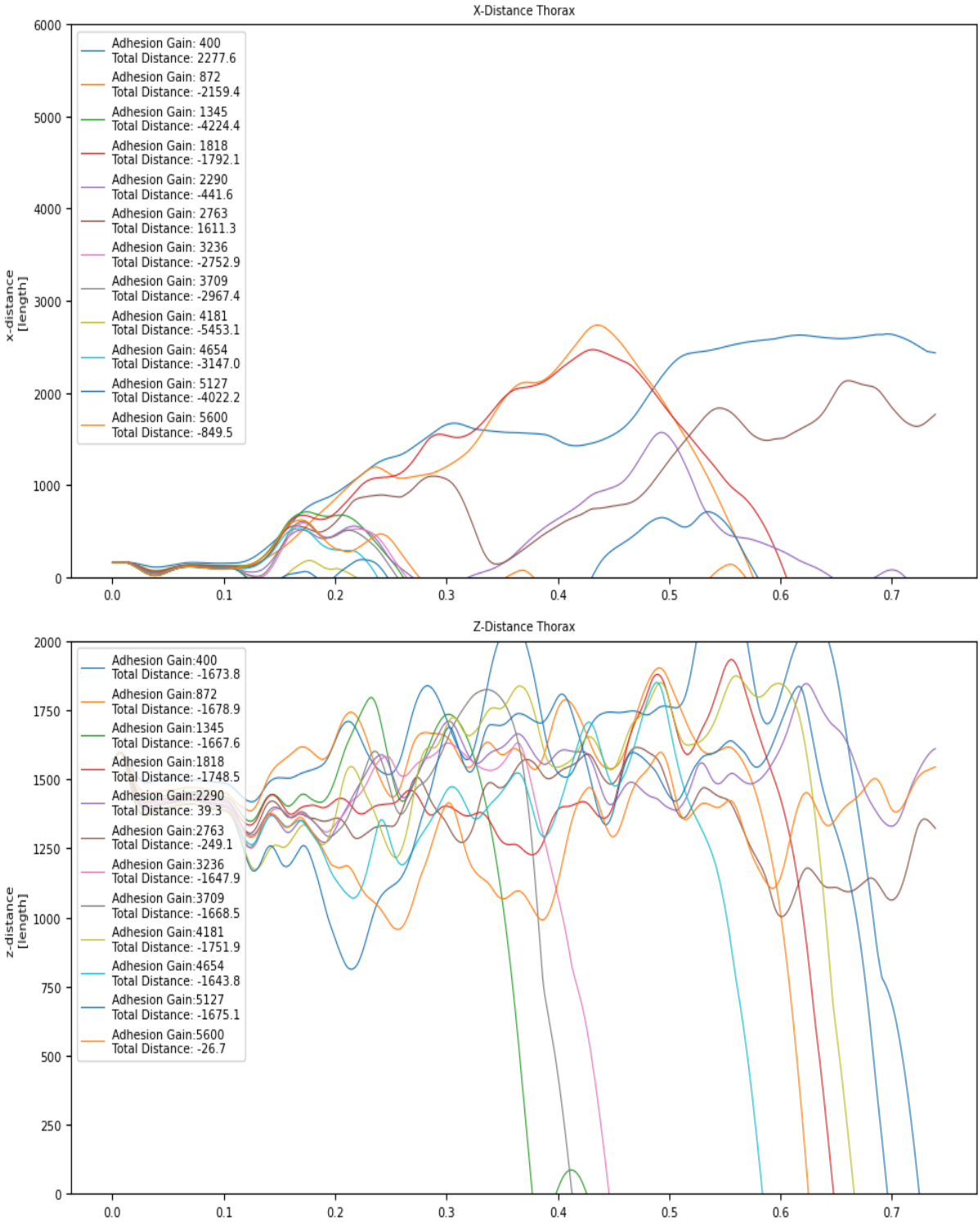

Figure 14: Distance travelled after calculating the adhesion vector during the simulation with windowsize 501. The method was applied on different adhesion gains for walking with adhesion in the following conditions: Normal Gravity, adhesion rule: " > ".

7 Discussion

7.1 Static: Adhesion On/ OFF on all Legs, single Leg and tripod

7.1.1 Grid Search to find max gain: Single Leg

Finding the correct gain is relevant for several reasons:

1. *If the adhesion gain is too large, the tarsi gooes through the ground.*
2. *If the adhesion is not strong enough, the fly flies away when the gravity is inverted.*

The results from the single leg grid in Figure 3 clearly show that a critical gain for a single leg to adhere exists. Since the model itself is symmetric and the positioning of the legs is asymmetric (Figure 2), the variation in critical gains between both sides are likely due to the distance of the tarsus 5 from the center of mass of the fly at starting time. It would be interesting to create a measure of stability to confirm this.
If this hypothesis is correct, it could be interesting to vary the amount of adhesion added according to the position of each leg with respect to the center of mass to increase the fly's stability.

Setting the adhesion gain to 4000 on the legs in tripod stance (i.e. on the left-front, left-hind and right-middle legs) for Figure 4 confirms that the adhesion strength is strong enough to pierce the floor. Similarly it shows that the leg does not reach exactly zero. This difference in height likely corresponds to the threshold set by FlyGym to detect a contact with the floor.

7.1.2 Behavior of Contact forces with adhesion

This experiment shows that the smoothed contact forces of the tarsus 5 have a higher variability than the sum of the contact forces of all tarsi (Figure 5, Figure 6). The sum of the contact forces of all tarsi allows to find the linear relationship between the contact forces and the adhesion force as suggested by the MuJoCo documentation [4]. This suggests that using the sum of all contact forces is more appropriate for the simulations. Nevertheless, only the tarsus 5's contact forces were used for the rest of the project, mainly for time reasons, as at first it was expected not to be a relevant distinction : when gravity is inverted, the other tarsi would not be in contact with the floor at all.

Adding adhesion to the tarsi also has other effects:

1. When adhesion is added or removed, the contact forces appear to overshoot.
2. If the interval at which adhesion is on / off is too short, the contact forces do not correct the overshooting, as there is not enough time.

3. However, less than 0.05s of the simulation is necessary to correct the overshooting.

These aspects are likely to be influenced by the window size used to smooth the contact forces. Increasing the window size further could potentially remove some of the variability. The kp values as well as the joint stiffness could also influence the adaptability of the simulation to the addition of adhesion.

The final gain was chosen to be the one enabling all single legs to be strong enough to adhere to the surface. Applying the maximum adhesion onto all legs, is not what logically happens with the real fly, hence the gain should be adjusted based on what leg is in contact with the ground, fly speed, etc... This has not been done because of the time constraints.

Furthermore it makes sense to expect that the adhesion control value should be different according to the direction of the gravity, either sideways, upwards or downwards.

7.2 Dynamic: Walking with adhesion

7.2.1 Prerun adhesion vector calculation

The " $>$ "-adhesion rule worked best for normal gravity walking amongst the two rules proposed (subsection 6.3). However for inverted gravity both rules could not overcome gravity. This is likely because the duration of both $\frac{dF_{contact}}{dt} > 0$ and $\frac{dF_{contact}}{dt} \geq 0$ does not coincide with the stance duration of the three legs required to be in stance for tripod gait walking. The lack in overlap, causes the D. Melanogaster to fly away when the legs are not in contact. 4016 in adhesion gain is sufficient for the fly to hold according to what was previously found. When gravity is inverted, we can hypothesise that the median filter does not allow the adhesion vector to adapt fast enough, as the fly will want to align its center of mass with the forces acting upon its body. Counteracting this impulse would likely require adhesion to be switched on for longer. An idea would be adding adhesion (with possibly a lower strength) for N-timesteps before its supposed activation according to the adhesion rules proposed above.

The iterative method also faces another issue. In the performed experiment, the adhesion vector calculated based on any simulation run with adhesion is influenced by the addition of the adhesion strength onto the contact forces. Since the adhesion strength is higher by a factor of at least 100, this influences the smoothing process and might negatively influence the adhesion vector calculated. It could therefore be useful to keep track of when adhesion is added and for example remove $0.8 \times (adhesion\ strength)$ in the contact forces.
Due to time constraints, the contact forces were not plotted for the grid search and thus were not

analyzed.

7.2.2 Inline adhesion vector calculation

The inline adhesion vector in its current form does not really work well since most of the flies fall to their back or fly away. Normally median filters take into account the values before and after the value that will be replaced by the median. When computing the medians during the simulation to later take the derivative, the median only takes into account past values. Because of this, the calculated median is not very representative of what will happen afterwards. To solve this problem it would be very useful to apply reinforcement learning to find the appropriate time points at which the adhesion should be turned on.

7.2.3 Further adhesion rules for future experiments

Currently, all adhesion rules depend on the derivative of the contact forces. As mentioned above, this is a problem when calculating the derivative during the simulation. For this, it could be interesting to set a specific threshold for each leg. If the contact force is higher than that threshold, adhesion should be applied. Adhesion can also be dependent on the relative position of the end effector compared to the center of mass. In later stages, changing the adhesion strength according to this position could be interesting.

8 Conclusion

The addition of adhesion has not yielded a smooth walking fly for any of the adhesion rules proposed. However many advances have been made. It is possible for the fly to adhere to the floor through only one leg. This is dependent on the positioning of the leg with respect to the body and its center of mass. The relationship between the contact forces and the adhesion strength was found to be linear, but requires an adaptation period by the system. Out of all proposed adhesion rules, the "¿"-adhesion rule performed best. It relatively well for normal gravity walking. None of the proposed methods worked for inverted gravity, however for further research the finetuning of the iterative and the inline / dynamic method of calculating the adhesion vector can be promising.

However it should be kept in mind that the data set used for this project was in parts manually adapted from joint angle data obtained from the fly walking on a ball. This is not necessarily representative of the fly walking on a smooth surface. Furthermore, it should be considered that during this project, the fly only walks on very smooth, flat terrain.

9 Gantt Chart

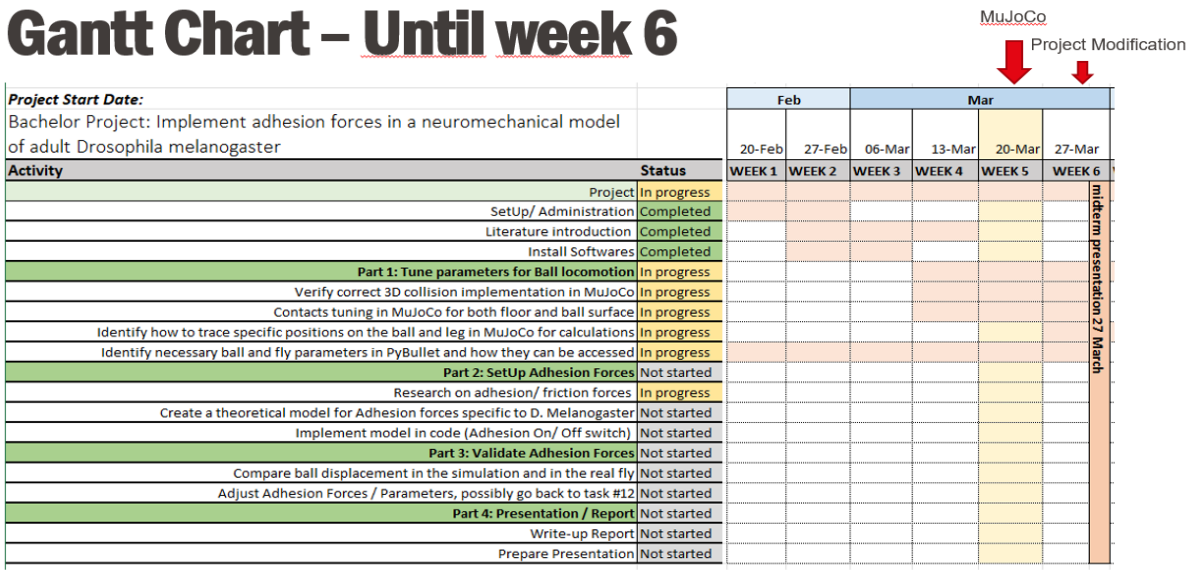

Figure 15: Original Gantt chart upto week 6. The lab was still transferring NeuroMechFly onto MuJoCo from Pybullet. For this reason I had planned several weeks of literary research. Starting to work with MuJoCo got delayed due to this.

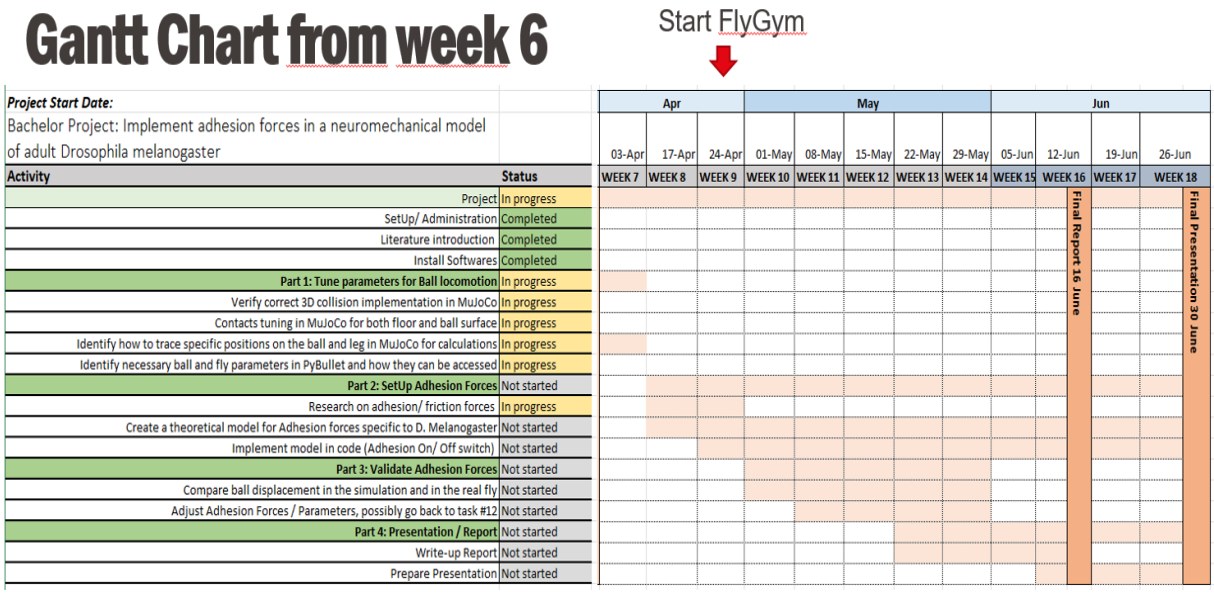

Figure 16: Original Gantt chart from week 6. The original project was to adapt parameters for the fly to walk on the ball, this however got modified during the midterm presentation. The Gantt chart is therefore no longer for this current project. Furthermore the lab was working on implementing FlyGym. This meant that I had to readapt my findings onto FlyGym after week 9. FlyGym was a great tool for my project, however it took me quite some time to understand how to implement adhesion in a manner that is coherent with the reinforcement learning aspect of FlyGym.

References

[1] V. Lobato-Rios, S. T. Ramalingasetty, P. G. Özdil, J. Arreguit, A. J. Ijspeert, and P. Ramdya, "NeuroMechFly, a neuromechanical model of adult Drosophila melanogaster," Nat. Methods, vol. 19, no. 5, Art. no. 5, May 2022, doi: 10.1038/s41592-022-01466-7.

[2] B. D. DeAngelis, J. A. Zavatone-Veth, and D. A. Clark, "The manifold structure of limb coordination in walking Drosophila," eLife, vol. 8, p. e46409, Jun. 2019, doi: 10.7554/eLife.46409.

[3] Ramdya et al., "Climbing favours the tripod gait over alternative faster insect gaits," Nat. Commun., vol. 8, no. 1, Art. no. 1, Feb. 2017, doi: 10.1038/ncomms14494.

[4] "XML Reference - MuJoCo Documentation." https://mujoco.readthedocs.io/en/stable/XMLreference.html (accessed Jun. 11, 2023).

[5] "`dm_control`: Software and Tasks for Continuous Control." Accessed June 11, 2023. https://www.deepmind.com/publications/dm-control-software-and-tasks-for-continuous-control.

[6] "NeuroMechFly." Python. 2020. Reprint, Neuroengineering Laboratory @EPFL - Ramdya Lab, March 10, 2023. https://github.com/NeLy-EPFL/NeuroMechFly

[7] "NeLy-EPFL/flygym." Neuroengineering Laboratory @EPFL - Ramdya Lab, Apr. 25, 2023. Accessed: Jun. 11, 2023. [Online]. Available: https://github.com/NeLy-EPFL/flygym

[8] Introduction Drosophila - a model system ... : Tolwinski, Nicholas. "Introduction: Drosophila—A Model System for Developmental Biology." Journal of Developmental Biology 5, no. 3 (September 20, 2017): 9. https://doi.org/10.3390/jdb5030009

[9] Hüsken, Mirko, Kim Hufnagel, Katharina Mende, Esther Appel, Heiko Meyer, Henrik Peisker, Markus Tögel, et al. "Adhesive Pad Differentiation in Drosophila Melanog

[10] Miller, Conrad. "Drosophila Melanogaster." Animal Diversity Web. Accessed June 13, 2023. https://animaldiversity.org/accounts/Drosophila_melanogaster/.

10 Appendix

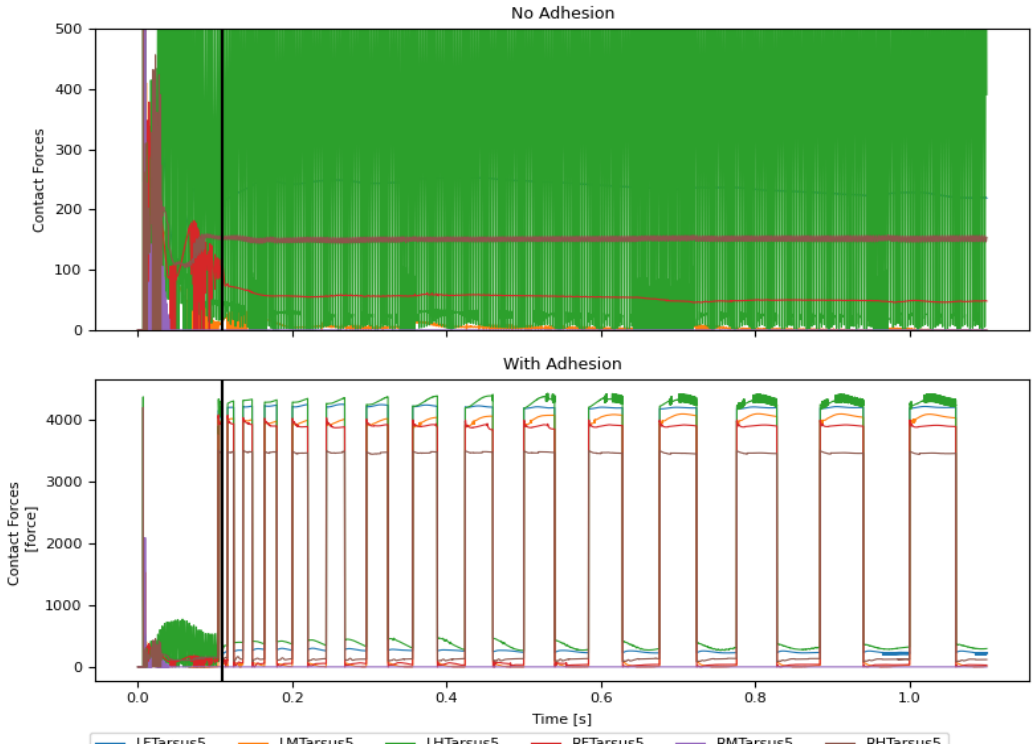

Figure 17: Comparison of static fly adapting to the addition of adhesion. The adhesion gain is 4000. The contact forces are those of only the Tarsi5.

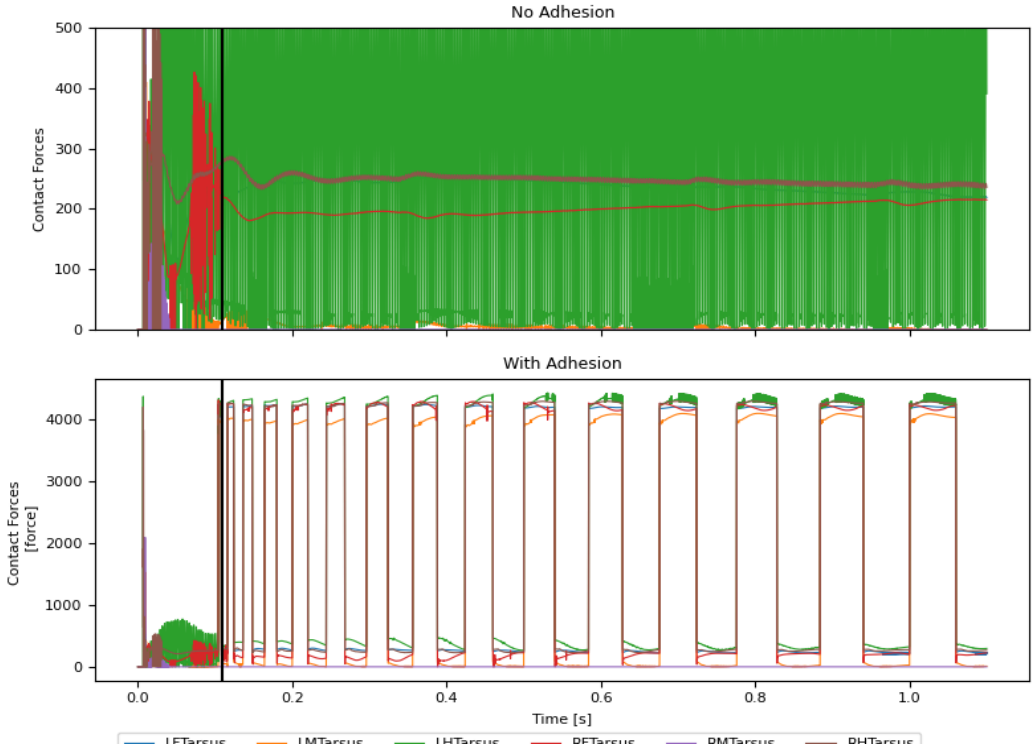

Figure 18: Comparison of static fly adapting to the addition of adhesion. The adhesion gain is 4000. The contact forces are the sum of all Tarsi5 contact forces.

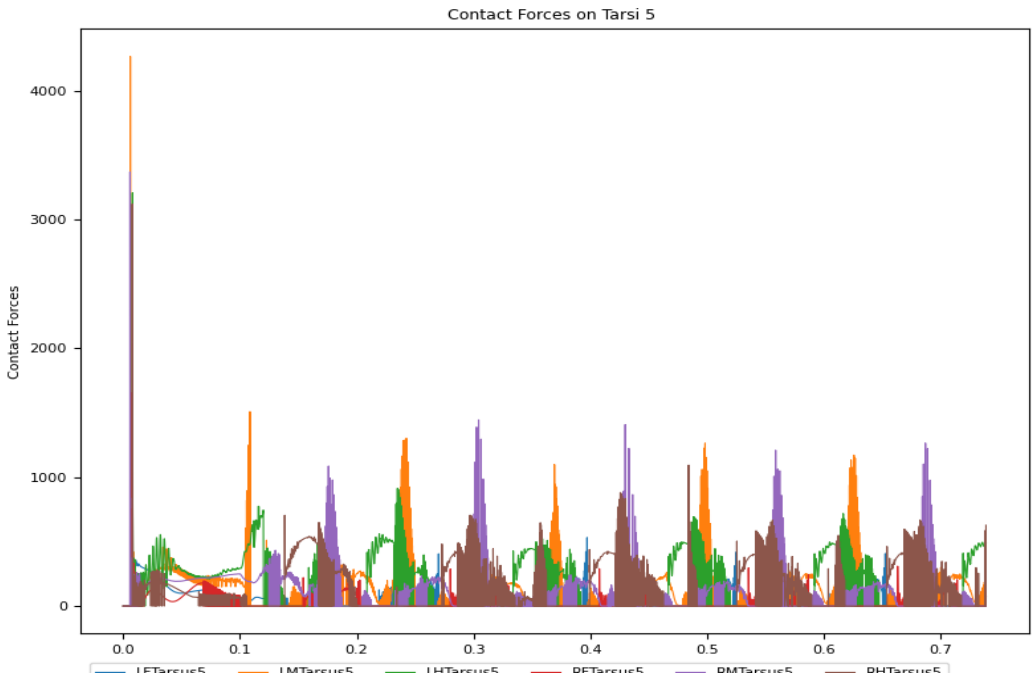

Figure 19: The contact forces are those of only the Tarsi5 when the fly is walking without adhesion at normal gravity.

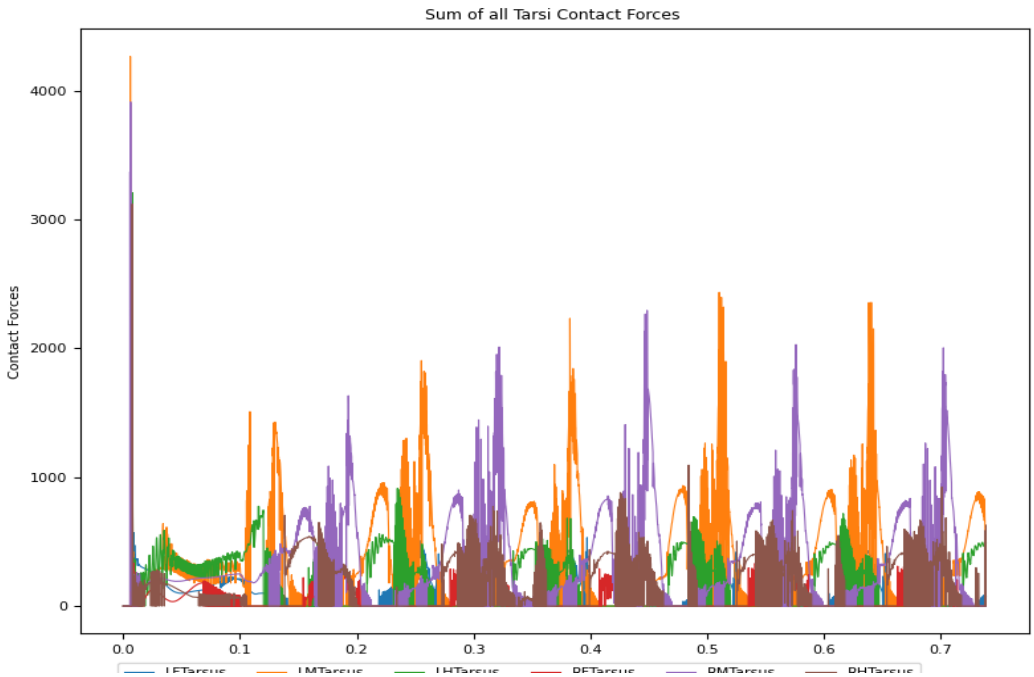

Figure 20: The contact forces are the sum of the contact forces of all tarsi when the fly is walking without adhesion at normal gravity. The data is filtered with a windowsize of 501.

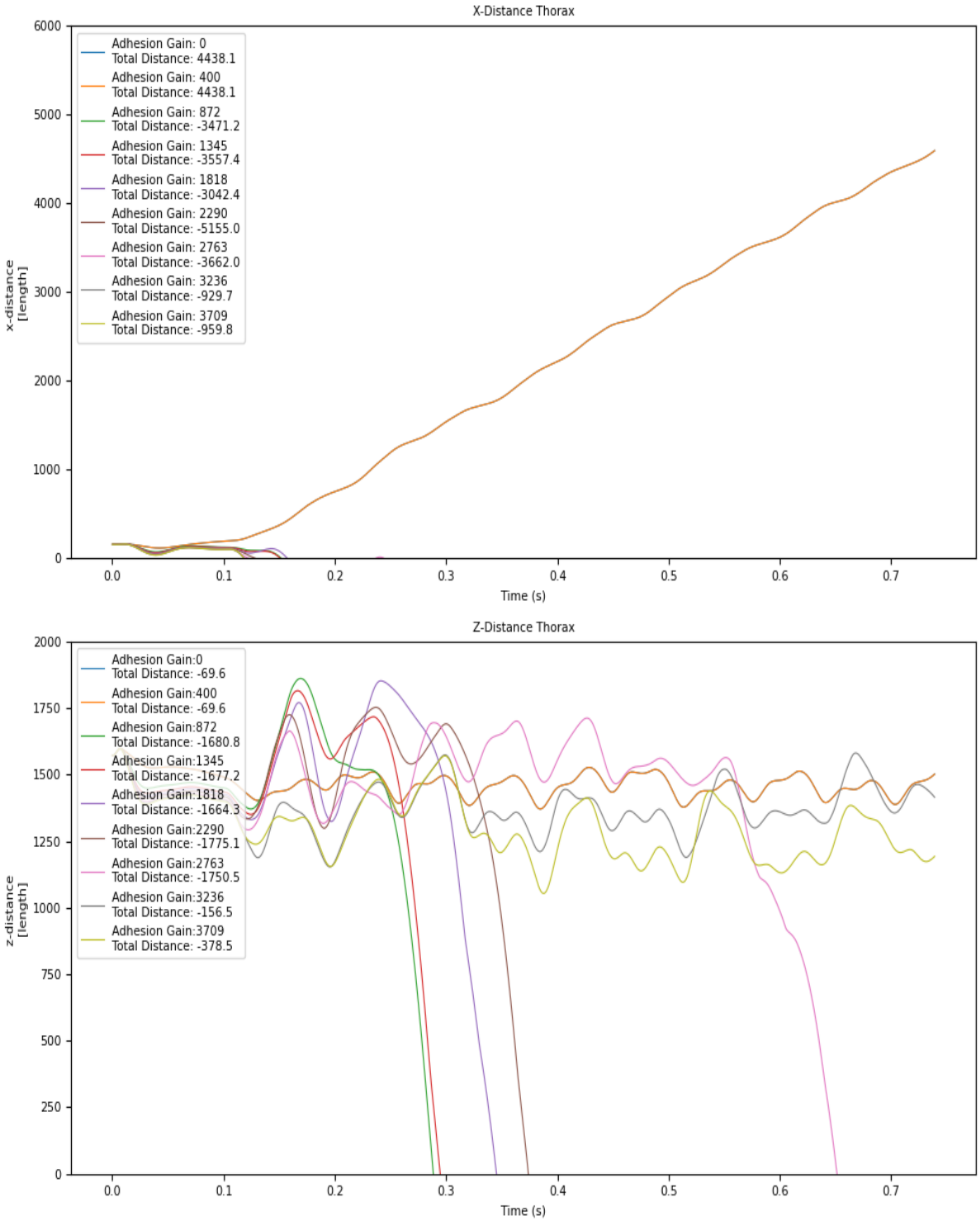

Figure 21: Distances travelled after using the simple method on different gains for walking with adhesion using the following conditions: Normal Gravity, adhesion rule: " $\geq$ ".

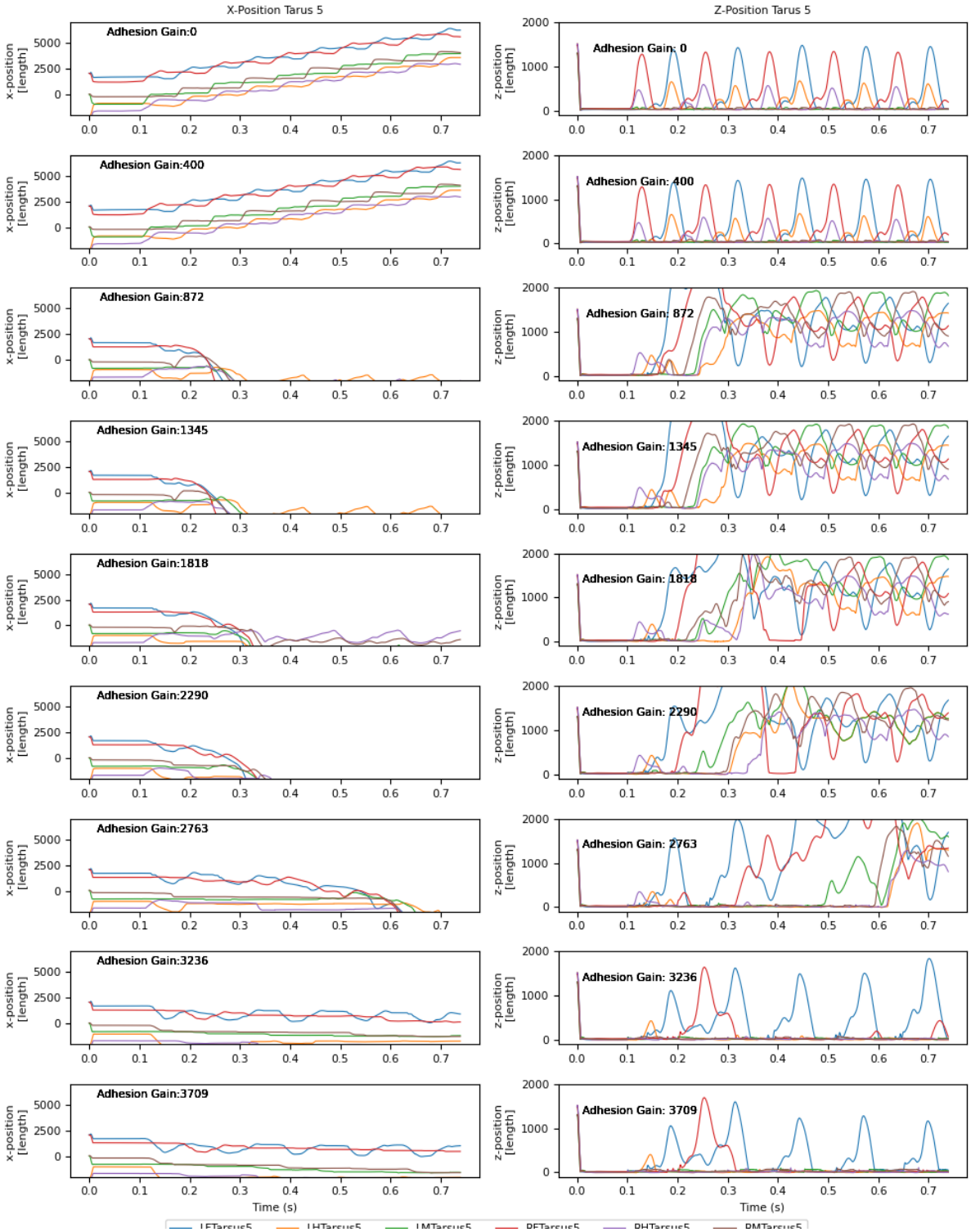

Figure 22: End effector positions travelled after using the simple method on different gains for walking with adhesion using the following conditions: Normal Gravity, adhesion rule: " $\geq$ ".

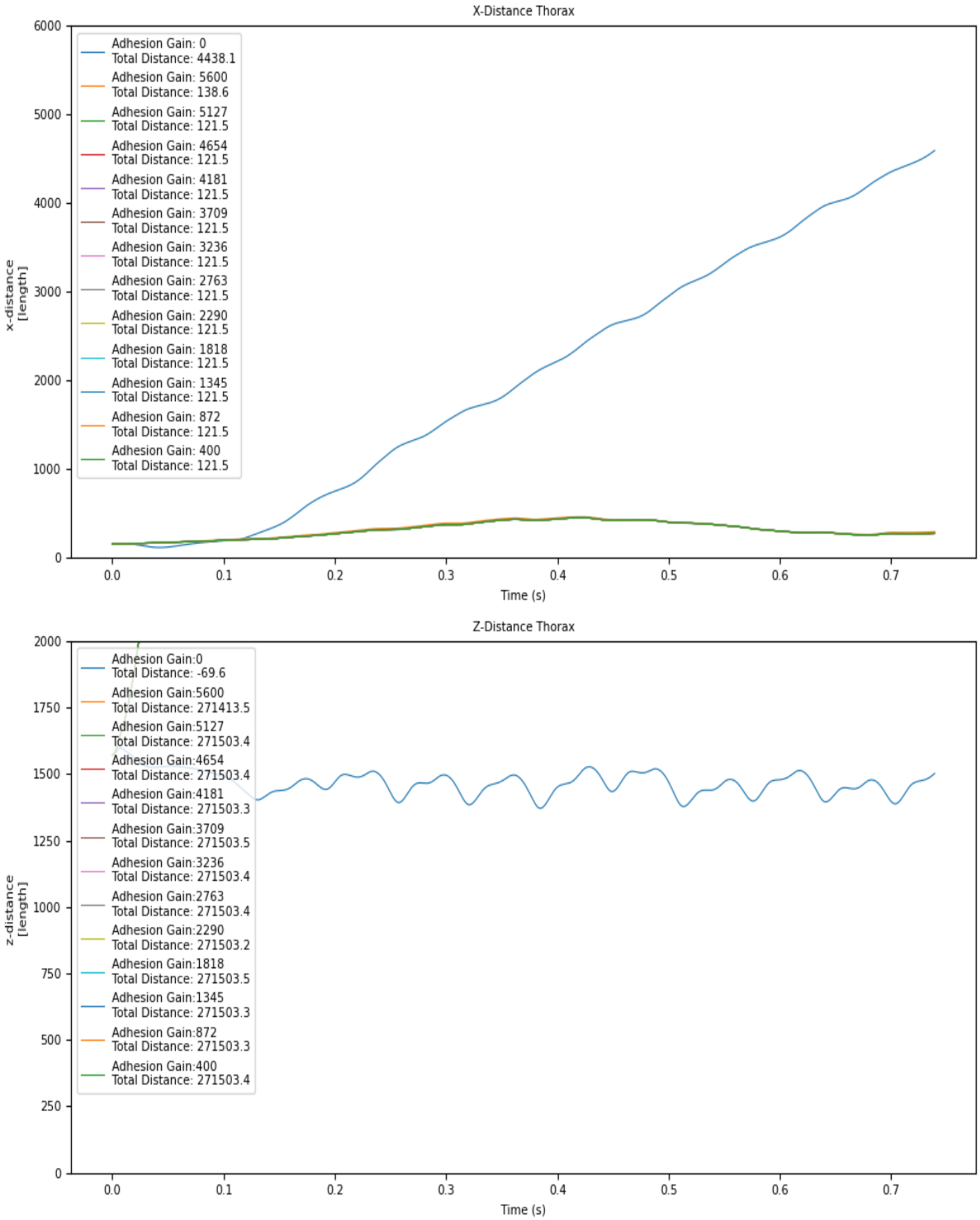

Figure 23: Distance travelled after using the simple method on different gains for walking with adhesion using the following conditions: Inverted Gravity, adhesion rule: " > ". The end effector positions of the normal gravity walking without adhesion is included for reference.

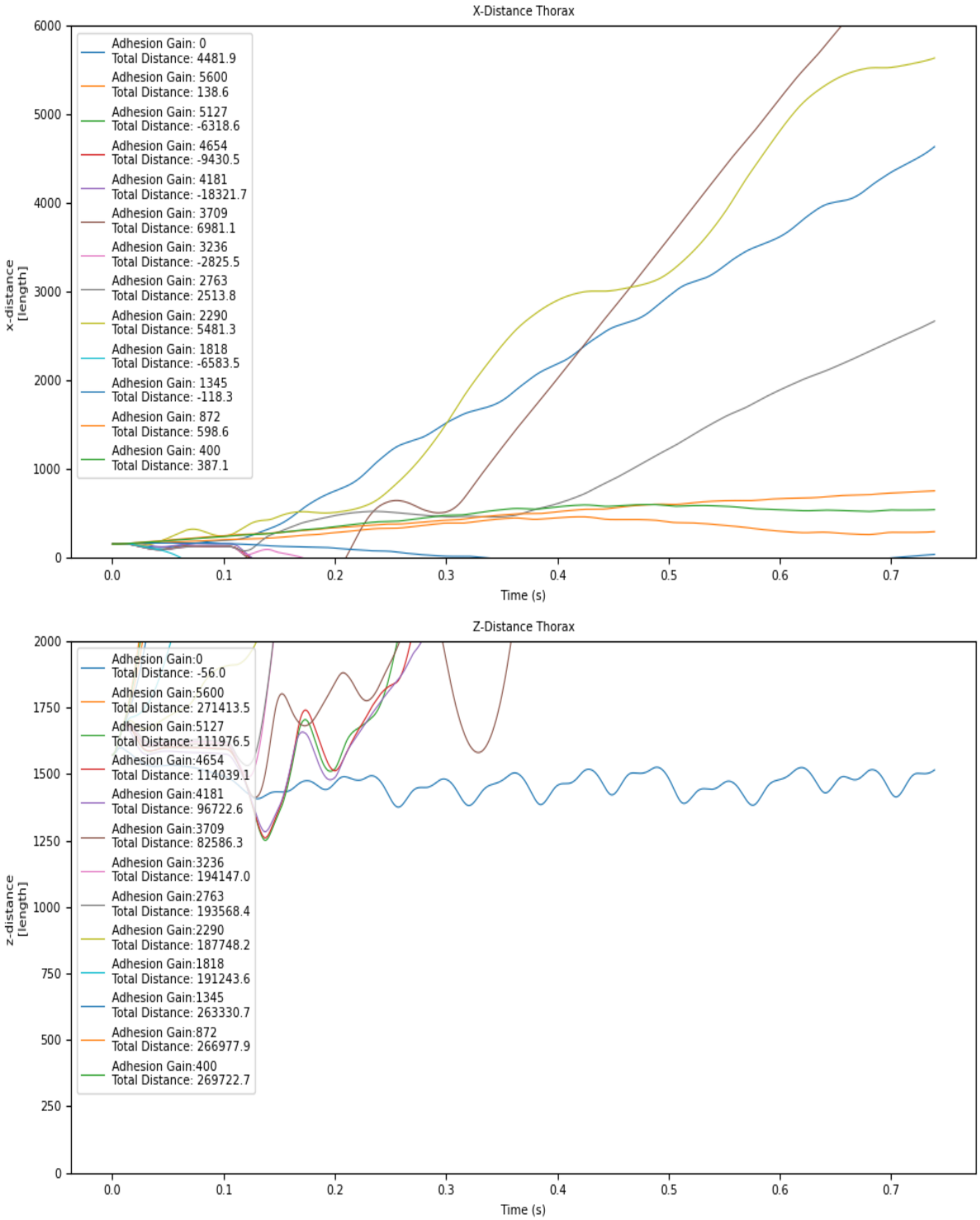

Figure 24: Distance travelled after using the simple method on different gains for walking with adhesion using the following conditions: Inverted Gravity, adhesion rule: " $\geq$ ". The end effector positions of the normal gravity walking without adhesion is included for reference.

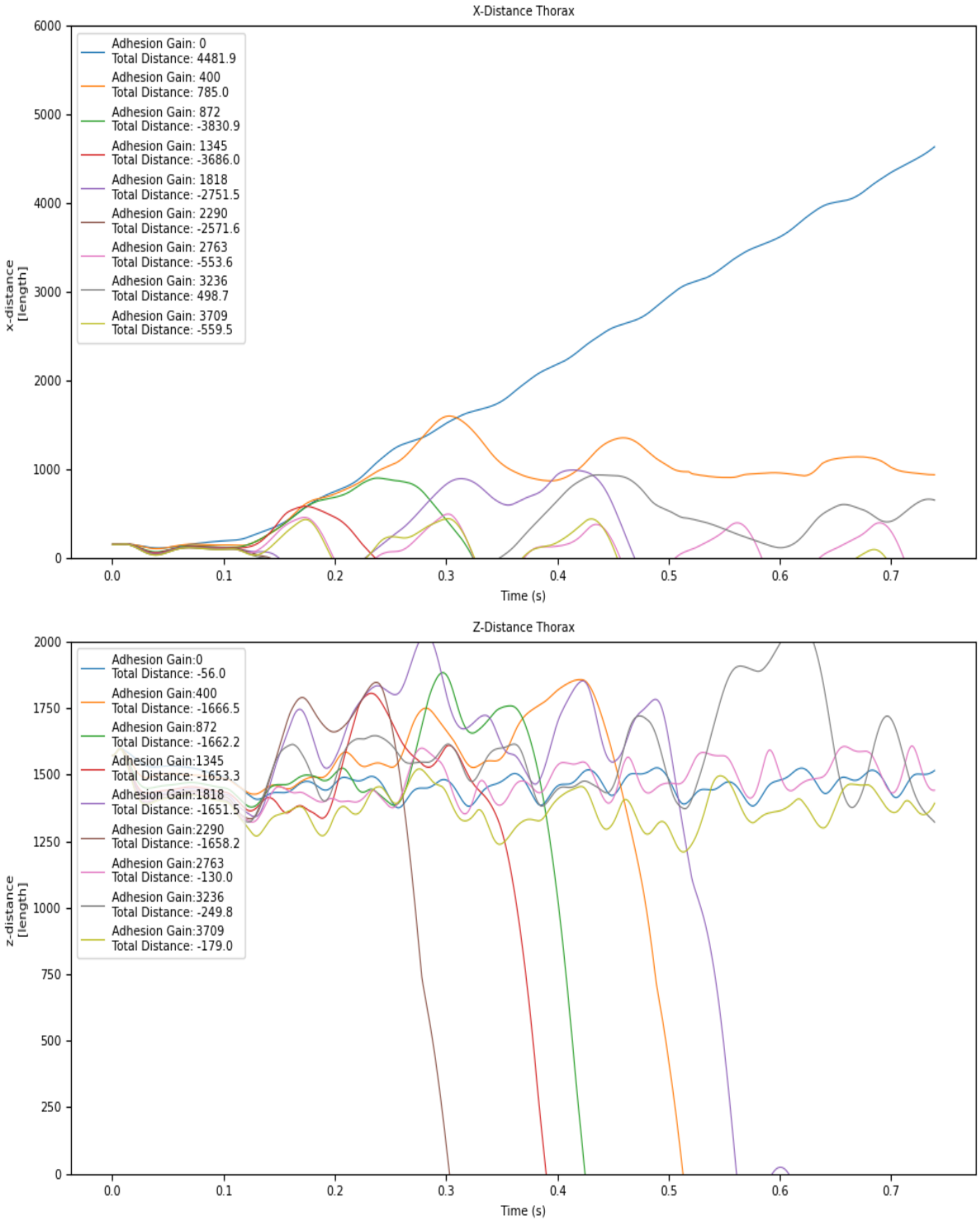

Figure 25: Distances travelled after using the Iterative method on different gains for walking with adhesion using the following conditions: Normal Gravity, adhesion rule: " $\geq$ ".

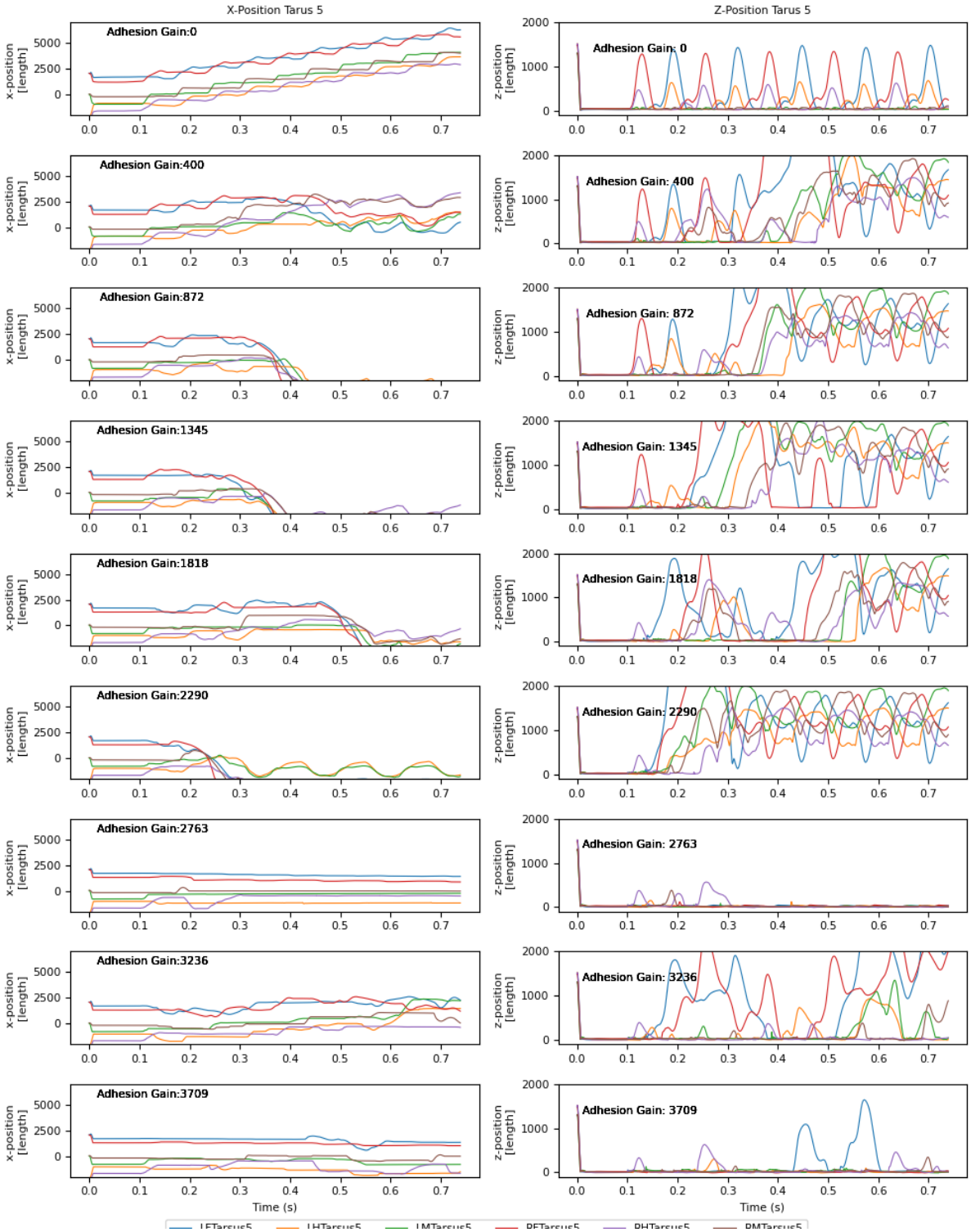

Figure 26: End effector positions travelled after using the Iterative method on different gains for walking with adhesion using the following conditions: Normal Gravity, adhesion rule: " $\geq$ ".

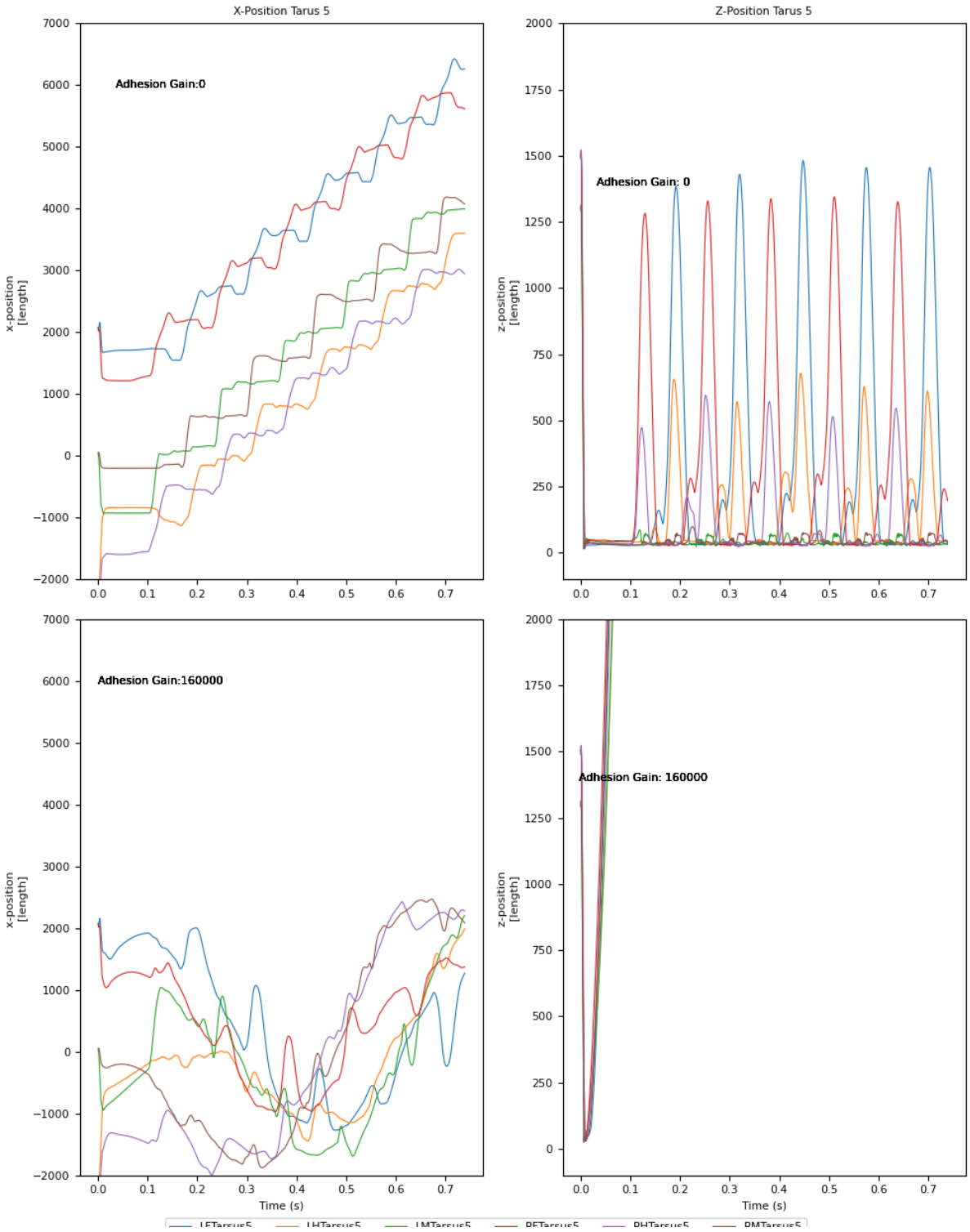

Figure 27: End effector positions travelled after using the Iterative method on different gains for walking with adhesion using the following conditions: Inverted Gravity, adhesion rule: " > ". The end effector positions of the normal gravity walking without adhesion is included for reference.

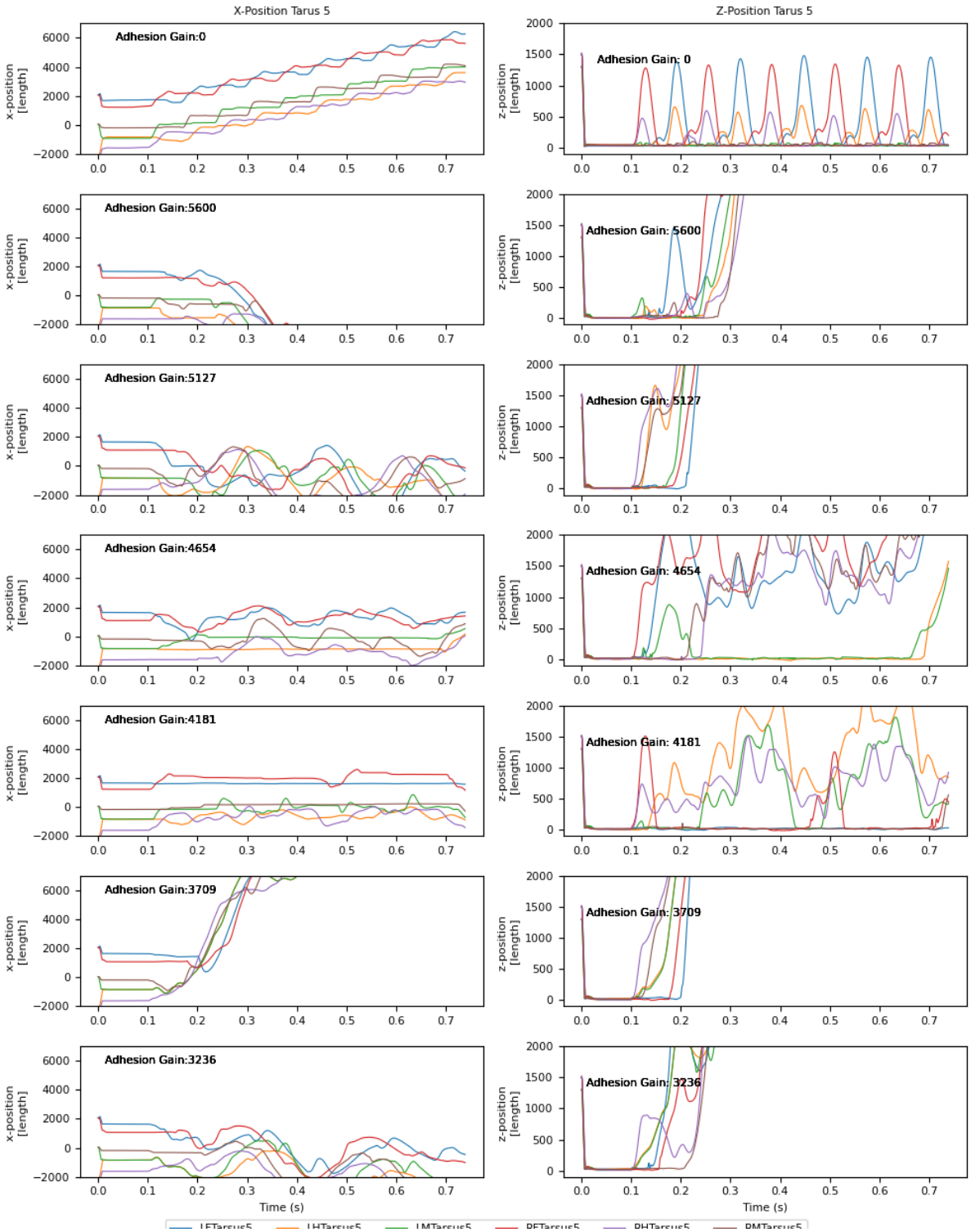

Figure 28: End effector positions travelled after using the Iterative method on different gains for walking with adhesion using the following conditions: Inverted Gravity, adhesion rule: " $\geq$ ". The end effector positions of the normal gravity walking without adhesion is included for reference.

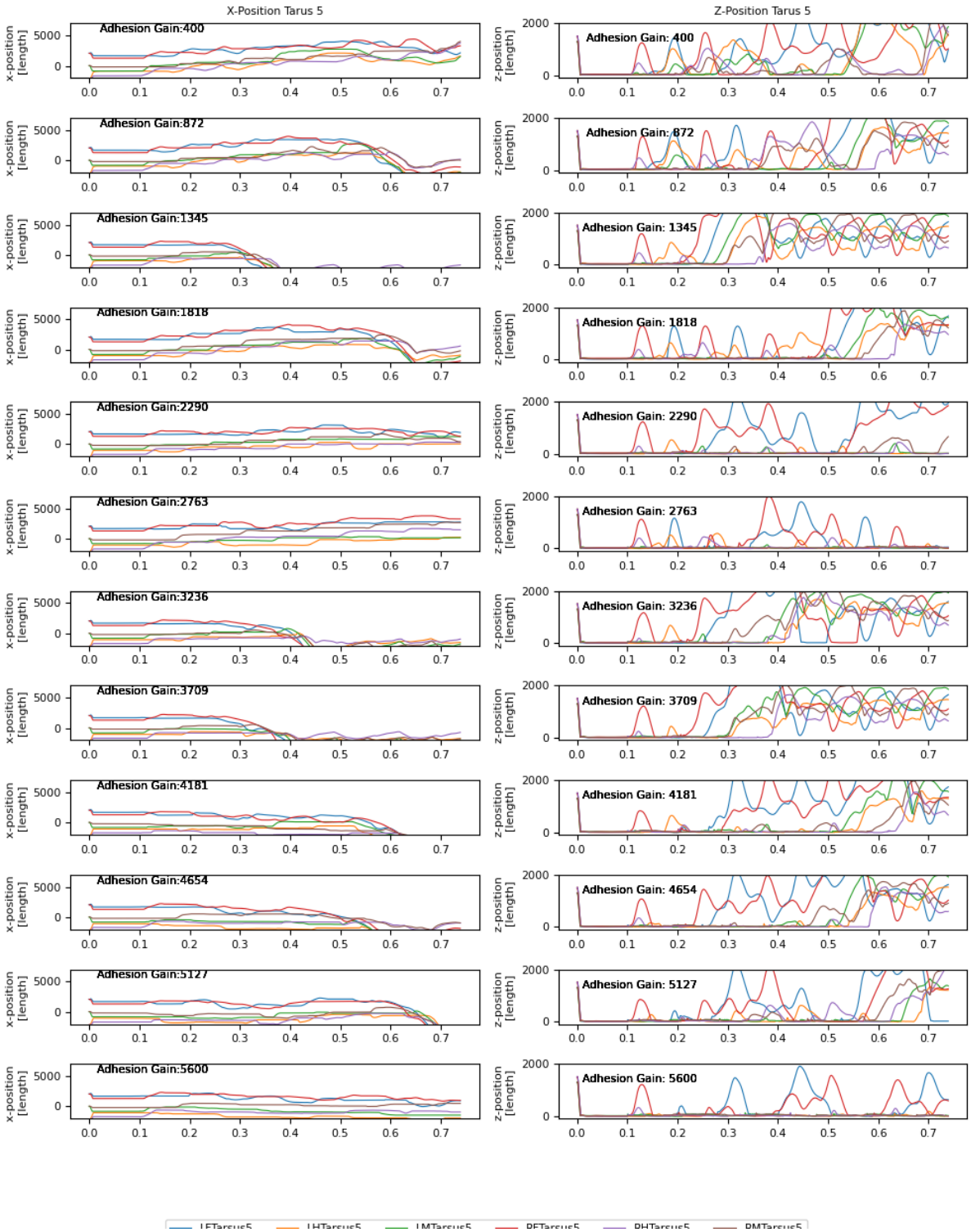

Figure 29: End effector positions travelled after calculating the adhesion vector during the simulation with windowsize 501. The method was applied on different adhesion gains for walking with adhesion in the following conditions: Normal Gravity, adhesion rule: " $\geq$ ".

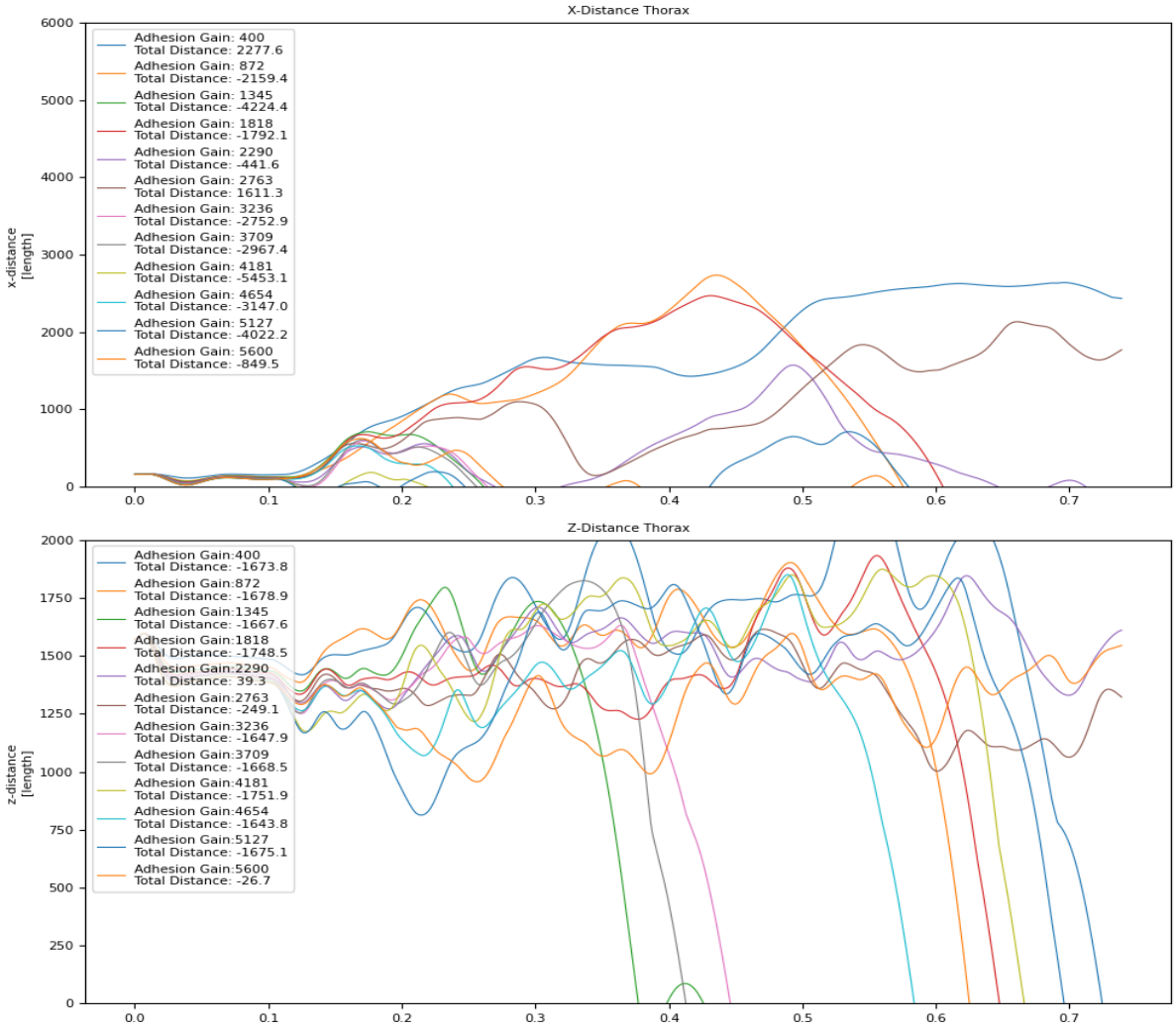

Figure 30: Distance travelled after calculating the adhesion vector during the simulation with windowsize 501. The method was applied on different adhesion gains for walking with adhesion in the following conditions: Normal Gravity, adhesion rule: " $\geq$ ".

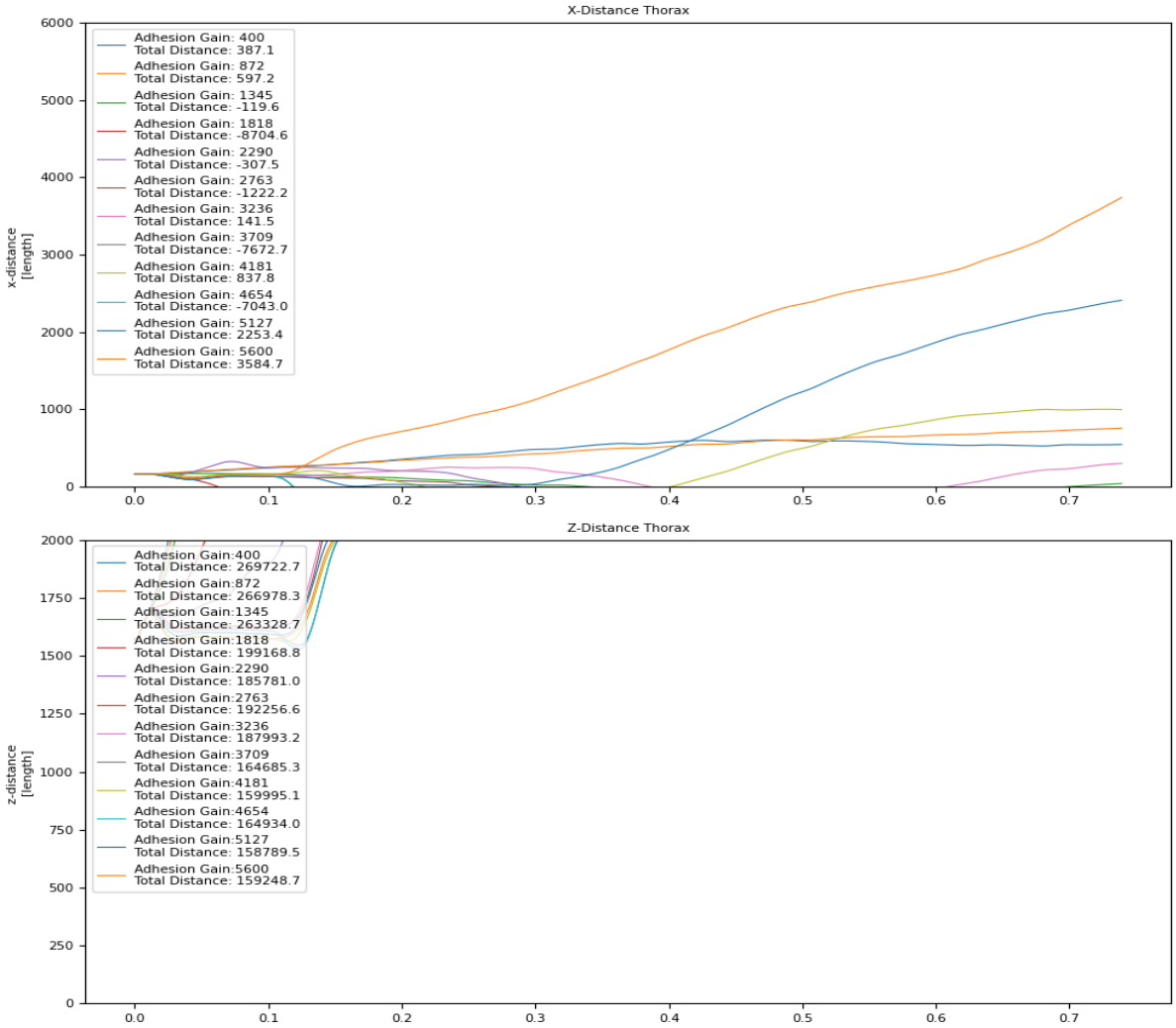

Figure 31: Distance travelled after calculating the adhesion vector during the simulation with windowsize 501. The method was applied on different adhesion gains for walking with adhesion in the following conditions: Inverted Gravity, adhesion rule: " > ".

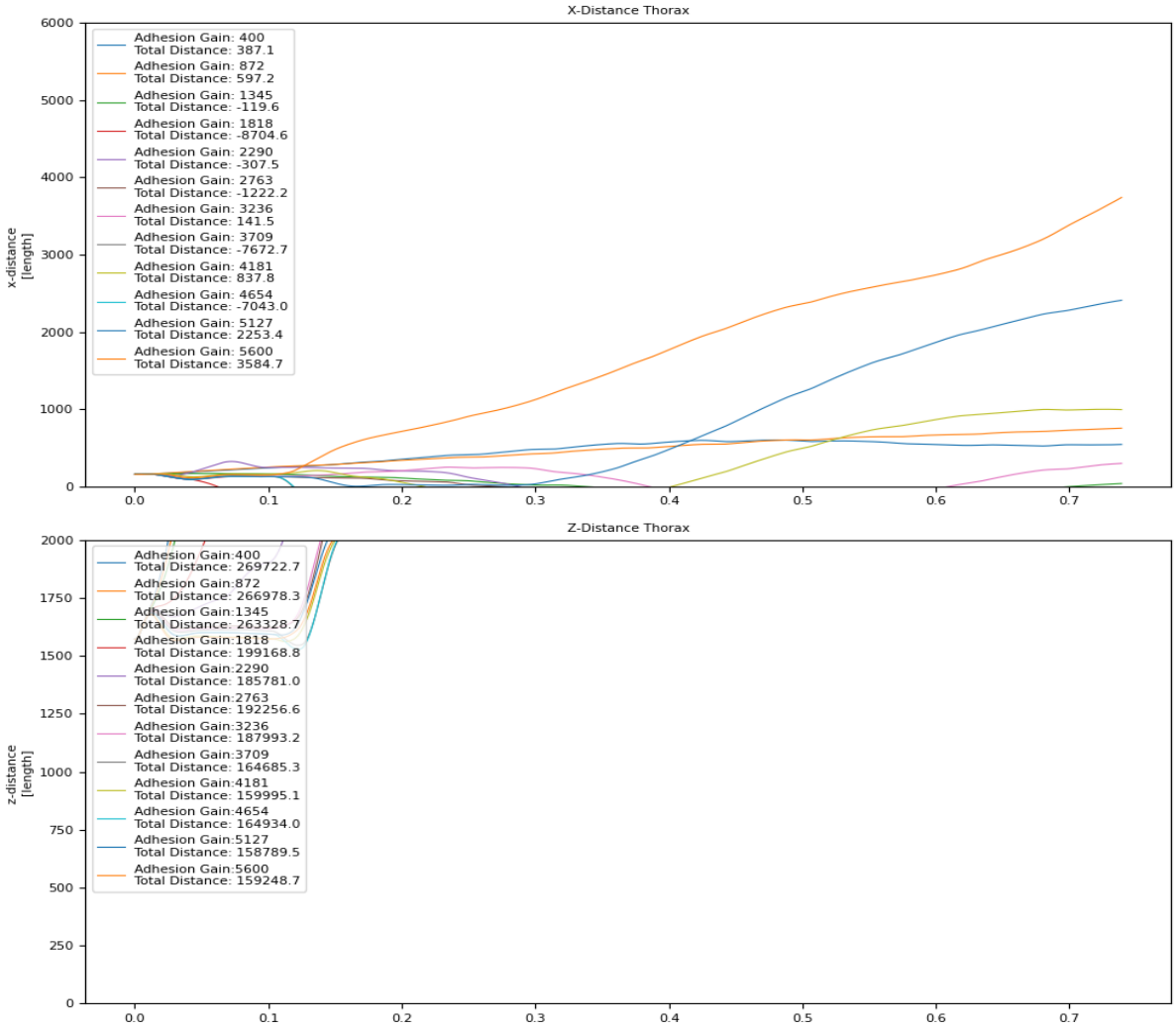

Figure 32: Distance travelled after calculating the adhesion vector during the simulation with windowsize 501. The method was applied on different adhesion gains for walking with adhesion in the following conditions: Inverted Gravity, adhesion rule: " $\geq$ ".