

Technical Report: Measuring fidelity of DTs Robotic Arm

Paula Muñoz, Javier Troya, Antonio Vallecillo
ITIS Software
University of Málaga (Spain)

Manuel Wimmer
CDL-Mint
Johannes Kepler University (Austria)

Contents

I	Arduino Braccio in the ITIS Software	1
I-A	System description	1
I-B	Scenarios	1
I-B1	Simple Moves	1
I-B2	Pick & Drop	2
I-C	Gap Tuning	3
I-C1	Simple Gap	4
I-C2	Affine Gap	5
I-D	Fidelity assessment	7
I-D1	Scenario Simple Moves	8
I-D2	Scenario Pick and drop	13
References		18

I Arduino Braccio in the ITIS Software

A. System description

The Atenea Research Group is a part of the ITIS Software Institute in Málaga. Their research is focused on applied research on modeling software systems, and recently, they started studying how to apply modeling concepts to Digital Twins. In this context, they developed a conceptual architecture and applied it to two cyber-physical systems: a Lego Mindstorms NXJ [1] and an Arduino Braccio [2].

Based on a set of requirements from a manufacturing plant and a set of engineering models, they developed a digital replica of an industrial robotic arm in USE [3]. This model is capable of simulating the passage of time and derives the servo positions based on the arm's speed.

To develop a Digital Twin System, they decided to find a robot whose behavior matched one of the set of engineering models. As a first proof of concept, they chose a basic Arduino Robotic Arm: The Arduino Braccio. This is a robotic arm kit developed by Arduino and RobotFactory (available at <https://store.arduino.cc/products/tinkerkit-braccio-robot>), offering an accessible platform for learning robotics and programming. The arm is composed of six servo motors, and its design allows for various movements and tasks, like object manipulation and drawing. It is controlled by an Arduino board with a dedicated software library, and users can program precise sequences of movements.

The research team developed a Digital Twin System (DT) based on these two systems, available at [4]. Now, they intend to use our tool to determine whether the robotic arm's behavior aligns faithfully with the set of engineering models or if they need to explore more advanced options.

B. Scenarios

Both twins operate using sequences of servo positions as input. By calibrating these sequences to the user's goals, the robotic arm can perform various behaviors. To test if this low-cost robot arm could fit its purpose, they provided us with information on two not-too-demanding scenarios.

1) Simple Moves

This first scenario includes a basic sequence of four commands performed at a low speed. The robotic arm starts in a position with all the servos extended and the gripper facing the ceiling. Then, it starts the following sequence:

```
Position positions[] = {
    Position(90, 90, 180, 172, 90, 10),
    Position(90, 55, 170, 86, 90, 10),
    Position(0, 90, 90, 90, 90, 43),
    Position(90, 90, 90, 90, 90, 73)
};
```

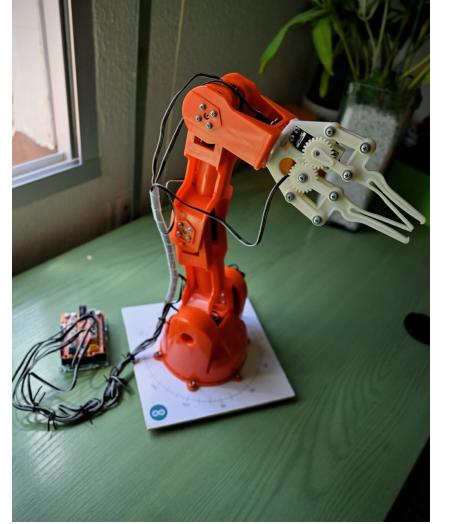


Fig. 1: Arduino Braccio in the ITIS Software Málaga

2) Pick & Drop

This second scenario includes a more complex sequence of three movements performed at a higher speed. The robotic arm starts in a position with all the servos extended and the gripper facing the ceiling. Then, the arm grasps a distant object, raises it 30 cm, moves it 50 cm to its right, lowers the arm, releases the object, and returns to the initial position. The commands that describe this sequence are:

```
#define GRIPPER_OPEN 10
#define GRIPPER_CLOSED 73

Position positions[] = {
    Position(90, 90, 180, 172, 90, GRIPPER_OPEN),
    Position(90, 90, 180, 172, 90, GRIPPER_CLOSED),
    Position(90, 55, 170, 86, 90, GRIPPER_CLOSED),
    Position(90, 90, 90, 90, 90, GRIPPER_CLOSED),
    Position(0, 55, 170, 86, 90, GRIPPER_CLOSED),
    Position(0, 55, 170, 86, 90, GRIPPER_OPEN),
};
```

C. Gap Tuning

Verifiability: The analysis performed in this section is available at https://github.com/atenearesearchgroup/fidelity-measure-for-dts/blob/main/src/evaluation/braccio/braccio_gap_tuning.ipynb

One of the main configuration parameters for alignment algorithms is the scoring system. In order to discern among the many possible alignments between two sequences, it is necessary to specify to the algorithm which decisions to prioritize when aligning the sequences. The classical Needleman-Wunsch algorithm, on which our proposal is based, has two configurable penalties (mismatch and gap) and one reward (match). These values can be assigned based on an input matrix that prioritizes certain characters over others (in the original approach for protein sequence alignment) or through fixed values.

One of the most common configurations when using fixed values is to assign +1 for a match, 0 for a mismatch, and -1 for a gap. This type of configuration prioritizes mismatches over gaps, favoring solutions with fewer gaps. In our case, we adapted this scoring configuration:

- **Match:** Value in the range (0, 1]. The more similar the snapshots are, the closer the value is to 1, based on the comparison function.
- **Mismatch:** Neutral penalty, 0. This occurs when the two snapshots fall outside the range of the Maximum Acceptable Difference (MAD). However, the algorithm considers that these snapshots should have matched for the optimal alignment.
- **Gap:** Negative penalty, aimed at prioritizing mismatches over gaps. This represents a state that is absent in the other trace.

This negative penalty can be configured in two ways:

- **Simple gap:** A fixed penalty that is added to the score each time a gap is included in the alignment.
- **Affine gap:** A configuration with two fixed penalties: one penalty for initiating a gap (P_{op}) and another usually smaller penalty for extending a previously initiated gap (P_{ex}).

The first approach produces alignments in which single-position gaps and matches alternate in the sequences. However, this alignment scheme may be less effective when the objective is to identify periods of anomalous behavior, as it tends to result in alignments with intermittent gaps in the trace. Conversely, in the second approach, we introduce penalties for such alignments and instead prioritize alignments where gaps are grouped together. Longer gaps facilitate the identification of anomalies, resulting in more meaningful alignments.

However, the latter approach demands more processing space and computational capacity. It not only requires one matrix to align the sequences using Dynamic Programming but also necessitates two additional matrices to evaluate whether to insert a gap or not in each of the sequences. Hence, in our algorithm, we incorporated the flexibility to configure alignments using both of these techniques. Depending on the specific scenario and the importance given to resource optimization, the user can select either approach. To assess the optimal configurations for penalties and their impact on alignment, we prepared experimental datasets for which we analyze the fidelity metrics introduced in Section III of the General Concepts Technical Report [5].

The configurations for the experiments conducted with the Robotic Arm are as follows:

Parameter	Range	Increments
Maximum Acceptable Distance (MAD)	[0.2, 4.7]	0.5
Penalty opening a gap (P_{op})	[-3.0, 0.0]	0.50
Penalty extending a gap (P_{ex})	[-2.0, 0.0]	0.10

This resulted in an analysis of 1400 alignments applied to Scenario *SimpleMoves*. The chosen range of MAD values depends on the system's domain. In the following two sections, we will examine the effects of these configurations on various metrics using figures such as Figure 2a and 2b. In these figures, all subfigures share the x-axis, where each unit represents an alignment applied to the scenario. Depending on the input values (MAD, P_{op} , P_{ex}), we obtain alignments with different statistics. The shading in this and subsequent figures represent change points, dividing the values into groups based on the values' tendency.

1) Simple Gap

The statistics for the percentage of matched snapshots, Frèchet distance, and average Euclidean distance between aligned points are depicted in Figures 2a and 2b. These figures specifically focus on the 200 input configurations where the P_{op} value is set to 0, implying that the cost of opening and extending a gap is the same. The change point, which signifies a shift in the values, is observed at the peak of the percentage of aligned snapshots. This peak represents a significant increase of 10% (from 80% to 90%) in the number of aligned snapshots compared to the remaining samples.

Figure 2a presents the samples sorted along the x-axis based on the increasing percentage of matched snapshots. The shaded red area highlights the alignments that achieve the best snapshot percentage and distances (Frèchet and Euclidean). These shaded alignments include MAD values that are distributed across the entire range, indicating that the increase in the MAD value makes the alignment constraint more flexible, thereby increasing the percentage of matched snapshots. On the other hand, when considering P_{ex} , it becomes evident that satisfactory results are only obtained with values greater than -1.0. Values below this threshold imply that the algorithm struggles to provide enough flexibility to incorporate an adequate number of gaps to align snapshots, leading to unsatisfactory outcomes.

We can understand this more easily by looking at Figure 2b. Instead of arranging the samples based on the percentage of matched snapshots, they are now sorted in increasing order of the number of gaps in the alignment. In this case, we can observe a similar change point as in Figure 2a, a 10% increase of matches snapshots. Once a certain number of gaps is reached, the alignments become satisfactory. This happens again for values of P_{ex} greater than -1.0, which means that the penalty for introducing a gap is low enough for the algorithm to prioritize an alignment that includes an adequate number of gaps. These gaps help to characterize the behavior of our system in comparison to the simulation, enabling the inclusion of delays, for example.

To verify the statistical relevance of the input values in relation to the output values, we performed linear regressions that relate the input parameters (MAD, P_{op} , P_{ex}) to the percentage of aligned snapshots. The results of this analysis are available in Table I. In this table, we have the values for the analysis of all samples (Simple-All) and specifically for the red-colored segment with the most optimal values (Simple-Segment). The results for the three input parameters are the following:

- **MAD** has a significant influence on explaining the variability of the data across all samples, as indicated by the statistical relevance with p-values below 0.05 and a coefficient of 2.13. However, by examining Figure 2a, we can observe that the values are distributed throughout the entire range in an increasing fashion, following the same increasing tendency as the percentage of matched snapshots. This implies that increasing the MAD loosens the alignment restrictions, resulting in a higher percentage of matched snapshots.
- **P_{op}** value is kept at zero for all the samples. Therefore, its coefficient is 0.00, which means that it doesn't affect the percentage of matched snapshots.
- **P_{ex}** coefficients and p-values indicate that it is statistically relevant when considering all samples (p-value is 0.00) but irrelevant within the range of optimal values in the segment (the p-value is greater than 0.05). This means that modifying the value of P_{ex} within the appropriate range of values [-1.0, 0] has little impact on the percentage of aligned snapshots.

The conclusion is that we can achieve satisfactory results for this example with **P_{ex} values between [-1.0, 0]**, regardless of the specific variations within that range.

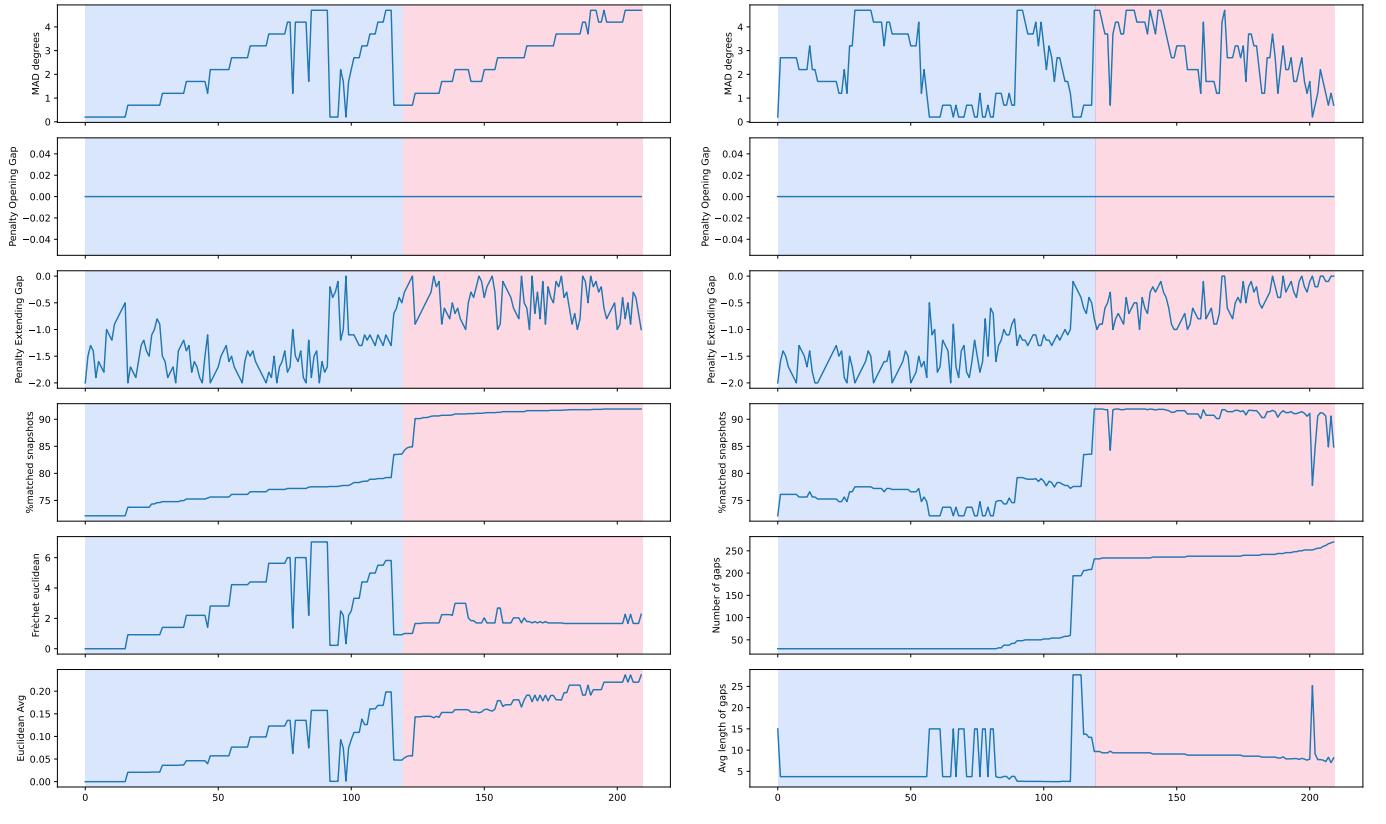


Fig. 2: Analysis of alignment statistics for the simple gap.

TABLE I: Analysis of the influence of gap penalty on the percentage of aligned snapshots.

Model	R-squared	F-statistic	Coef. MAD	p-value MAD	Coef. P_{op}	p-value P_{op}	Coef. P_{ex}	p-value P_{ex}
Simple-All	0.723	270.551	2.138 ± 0.196	0.000	0.000 ± 0.000	0.000	9.543 ± 0.465	0.000
Simple-Segment	0.884	314.988	0.411 ± 0.016	0.000	0.000 ± 0.000	0.000	0.117 ± 0.060	0.055
Affine-All	0.719	1070.941	2.137 ± 0.079	0.000	0.463 ± 0.133	0.001	9.297 ± 0.187	0.000
Affine-Segment	0.728	426.627	0.857 ± 0.027	0.000	0.493 ± 0.036	0.000	1.090 ± 0.105	0.000

2) Affine Gap

The analysis for the Affine Gap approach is similar to that performed for the simple approach. In Figures 3a and 3b, we have the same statistical analysis for the Affine Gap approach. The first plot displays the statistics sorted by % of matched snapshots, while the second plot sorts them by the number of gaps.

If we look at Figure 3a, the results are similar to those of the Simple Gap. MAD values in the segment with the highest percentage of alignments are evenly distributed throughout the range. The same happens for P_{op} , which shows that we can get satisfactory alignments for any value within the range of [-3, -0.5]. As for P_{ex} , the optimal values are obtained, just like in the previous case, within the range [-1.0, 0). The algorithm requires the gap costs not to be too high in order to include them and obtain relevant alignments.

Similarly, in Figure 3b, as we reduce the cost of P_{ex} , we increase the number of gaps that the algorithm adds to the alignment, allowing flexibility in the alignment choices and improving the number of aligned snapshots. We can also observe that as we reduce the cost of the gaps, the average length of the gaps decreases, prioritizing shorter gaps over longer ones.

Regarding the analysis of the statistical significance of the data, the results are in Table I (Affine-All, Affine-Segment) and are similar to the previous ones.

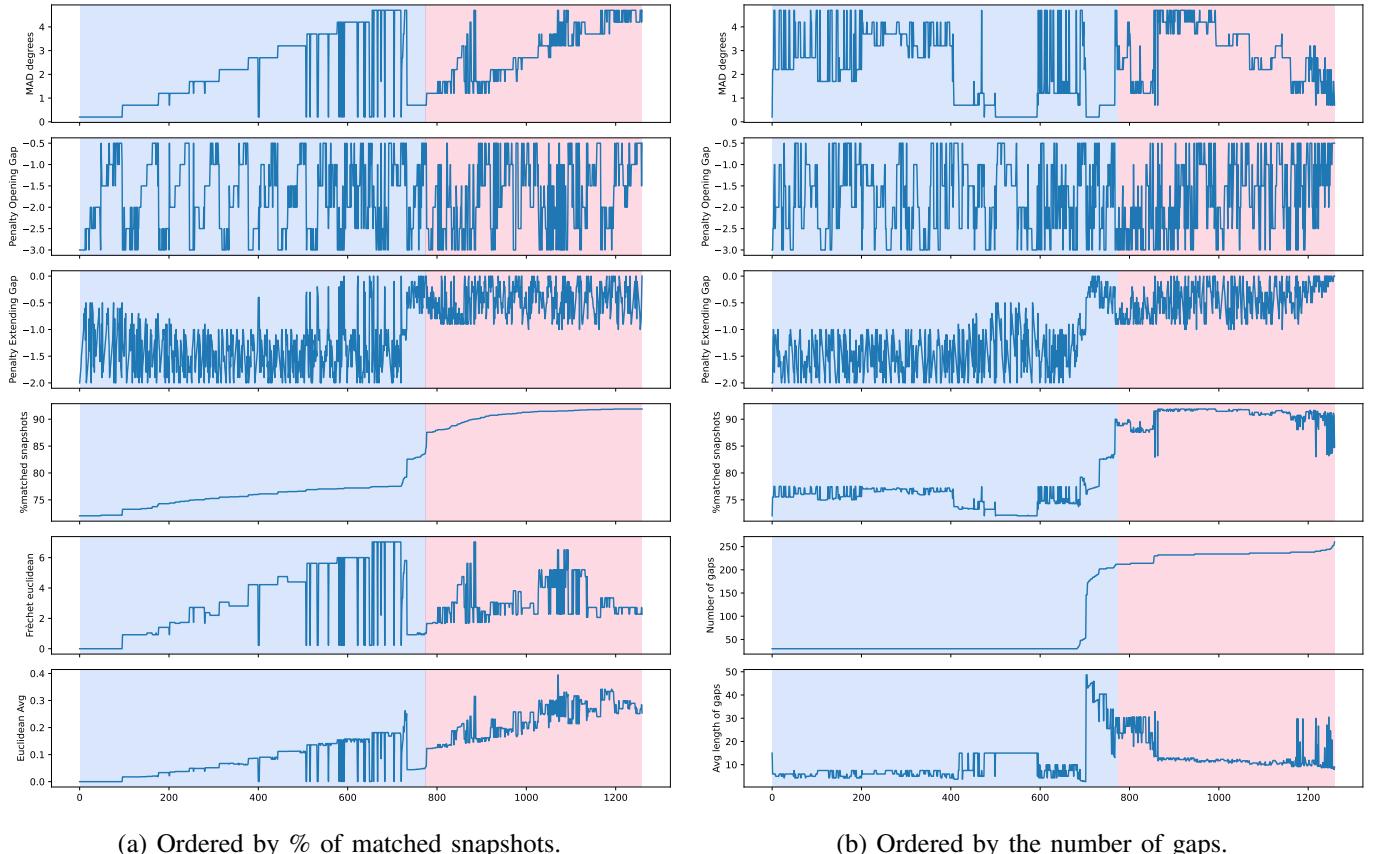


Fig. 3: Analysis of alignment statistics for the affine gap.

- **MAD** has a significant influence on explaining the variability of the data across all samples, as indicated by the statistical relevance with p-values below 0.05 and a coefficient of 2.13. However, by examining Figure 3a, we can observe that the values are distributed throughout the entire range in an increasing fashion, following the same increasing tendency as the percentage of matched snapshots. This implies that increasing the MAD loosens the alignment restrictions, resulting in a higher percentage of matched snapshots.
- **P_{op}** coefficients are really low for all samples, affecting around 0.4% for each unit we increase the penalty, which means that it barely affects the percentage of aligned snapshots.
- The **P_{ex}** coefficients are high considering all samples, as a change of 1 unit in P_{ex} would increase the percentage of aligned snapshots by 9%. However, when we only consider the aforementioned segment in which we obtain the optimal results, this coefficient is reduced to 1%. This indicates that modifying the value of P_{ex} within the appropriate range of values [-1, 0) has little impact on the percentage of aligned snapshots.

Therefore, the appropriate configurations for the algorithm would include a **P_{ex} value between [-1.0, 0)** and an **P_{op} value between (-3, 0)**. In our approach, we typically use the combination of **-1 as P_{op} and -0.1 as P_{ex}**, which is one of the recommendations from BLAST.

BLAST suggests that the penalty for initiating a gap should be 10 to 15 times higher than the penalty for extending it. The values for the penalties of opening and extending a gap for BLAST are obtained empirically and usually depend on the frequency scoring matrix used for the alignment [6]. However, generally, as a default value, the penalty for opening the score is approximately ten times higher than the cost for continuing a gap.

D. Fidelity assessment

Verifiability: The analysis performed in this section is available at https://github.com/atenearesearchgroup/fidelity-measure-for-dts/blob/main/src/evaluation/braccio/braccio_fidelity_analysis.ipynb

Next, we analyze the level of fidelity we achieve when aligning the simulator trace with the trace of the same behavior in the real system. For this analysis, we use some of the fidelity metrics we have defined in Section III of the General Concepts Technical Report [5]: the percentage of aligned snapshots, the Frèchet distance, and the average Euclidean distance (in the relevant area) between aligned snapshots.

To interpret these metrics, we need to consider what the values would be for the alignment of two identical traces. In that case, for any value of MAD:

- the **percentage of aligned points** would be 100%
- the **Frèchet distance** would be zero
- the **average Euclidean distance** would also be zero

These would be the results we would obtain for a model that had the maximum level of fidelity and was capable of accurately emulating the system. Anything that deviates from this model indicates a lower level of fidelity. We can compare different models and assess their fidelity level based on the metrics, using perfect alignment as a reference. In our work, we proposed a set of fidelity indicators to decide the degree of fidelity of a DT with respect to a PT depending on the values of the three metrics.

- **Alignment with %MS above 95% ($\pm 2\%$)**¹ is considered good enough, and the degree of fidelity depends on the distance between the traces.
- If **%MS is between 90% and 95% ($\pm 2\%$)**, alignment is low, but the distance metrics can be considered. The acceptable distance between the traces is application-dependent, and whether it is the Fréchet or the Euclidean distance that really matters.
- **If %MS below 90% ($\pm 2\%$):** traces could not be properly aligned, and therefore no faithful behavior can be expected.

The alignment algorithm applies the following configuration for all scenarios:

Parameter	Range	Increments
Maximum Acceptable Distance (MAD)	[0.2, 5.0 degrees / 18.0 mm]	0.2
Penalty opening a gap (P_{op})	-1	-
Penalty extending a gap (P_{ex})	-0.1	-

The specific and detailed guidelines on how to set the configuration values for Affine Gap are available in the previous section. In the incubator experiment, we do not consider any Low Complexity Areas. As for MAD, it was empirically established by determining where the plateau of fidelity metrics was achieved for illustrative purposes. To establish a single value in a practical example, we need to reason about the maximum distance we want to allow for aligning two snapshots. We will further develop this idea in the subsequent sections based on the data.

For each of the scenarios, we have two perspectives of the data. One is based on the movements expressed in terms of the angles of the servos, and the other is based on the grip's coordinates. We will compare the results of the two measurements and their effect on the fidelity metrics. This case study is particularly interesting compared to the previous ones, as we will, for the first time, deal with a system with multi-parameter snapshots and reason about how this aspect affects the alignments.

¹Note that we are assuming a maximum permissible error (MPE) of 2% [7] for the assessment of %MS, since most times thresholds are not completely accurate.

1) Scenario Simple Moves

The results for the three fidelity metrics for different MAD values are available in Table II and a visual representation of this data in Figure 4. The three metrics are available for both representations of the robot's movement: the angle of the six servo motors and the coordinates (x, y, z) of the arm's gripper. In both figures, we have the statistics arranged in ascending order by the value of the MAD.

As we can see, the trend of the metrics for the two representations is similar. It starts with a slow increase in the percentage of aligned snapshots until it reaches a plateau at approximately 91%. This plateau is reached at around 2 and 3 degrees in the case of the servo motor angles, which corresponds to two and three times its precision (one degree). If we analyze the plateau for the coordinates, we also find it between 10 and 15 mm, which is 2 and 3 times the precision (5 mm).

If we look at the distances, they also progressively increase as we raise the percentage of aligned snapshots, but this progression also diminishes upon reaching the plateau. Furthermore, in both cases, the Euclidean distance is below the precision of both sensors.

To analyze the alignment results more thoroughly, we will examine two of them at different levels of MAD. First, in Figure 5, we will take a MAD level of 0.8 degrees and 3 mm, which yields 85% of aligned snapshots. Upon closer inspection of the figure, we observe a series of aligned points and certain gaps in the curves. It is essential to consider that for a snapshot to be considered aligned, it must have a value within the assigned MAD range for all the attributes.

If we start with Figure 5a, we can observe that at the timestamp 8000, there is a large number of gaps. This is because, at this timestamp, several movements involving more than one servo coincide, and the distance of all of the combinations of snapshots attributes of the servos simultaneously moving cannot be aligned with the given MAD. In the rest of the transitions, not as many servos are involved, and the rest of them are in a stopped state, which makes it easier to align them.

To better understand this situation, let us take a look to the timestamp 8000 in Figure 6a, where we can see clearly that these simultaneous transitions result in a displacement of about 50 mm along the y-axis. In the simulation, the rise does not have the same slope, and the highest point does not exactly coincide. Therefore, we encounter various gaps, which are the ones we observe in the alignment in Figure 5a.

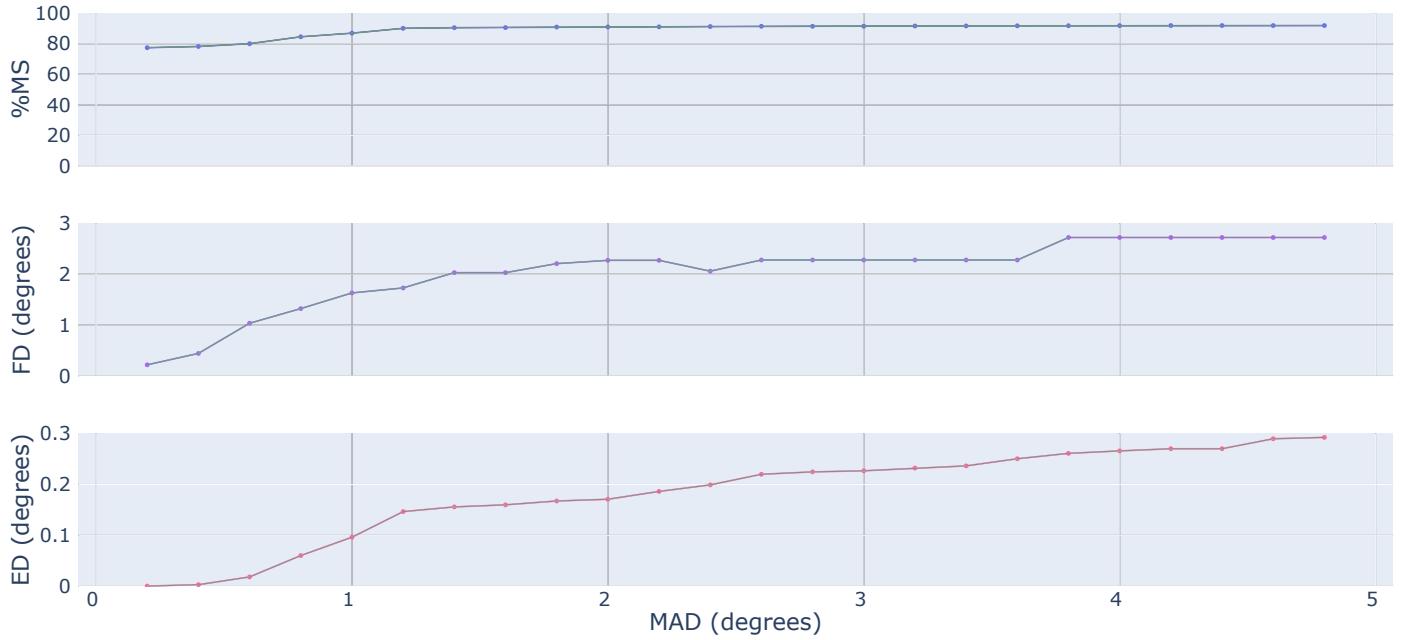
However, as we have seen in Figure 4, these MAD constraints are too strict, although they help us identify where the smallest differences between the traces are located. In Figure 6, we can already observe a satisfactory alignment of the previously mentioned parts. However, due to the small delay between the behaviors, we encounter 8% of gaps that prevent us from achieving a 100% alignment, along with a 1% of mismatches.

With this information, we can draw two conclusions:

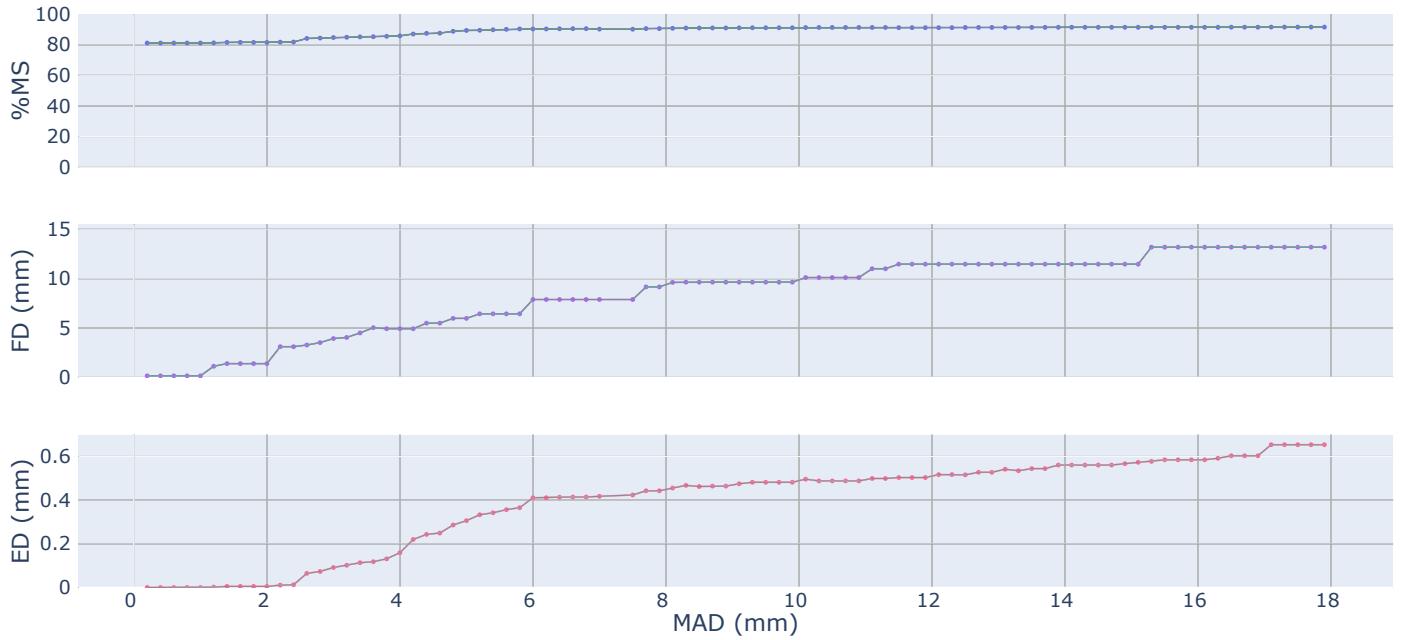
- The plateau in the metrics is reached at a **adequate MAD value that is approximately 2 or 3 times the accuracy of both measuring instruments**. Between 10 and 15 mm for the grip's coordinates and 2 or 3 degrees for the servo's axis angles.
- **The percentage of aligned snapshots is between 90 and 95% ($\pm 2\%$)**, which is enough to consider the distance metrics. Since the Euclidean distance is below the accuracy of the accelerometer, we could say that the Braccio Robot is faithful enough for this scenario.

TABLE II: Fidelity results for scenario *SimpleMoves*.

MAD	Axis position (degrees)			Grip's Coordinates (mm)			
	%MS	FD	ED	MAD	%MS	FD	ED
0.2	77.3918	0.2236	0.0007	3	84.7654	3.9309	0.0916
0.4	78.184	0.4472	0.0033	3.6	85.3138	5.0308	0.118
0.6	80.0122	1.0392	0.0186	4.2	87.081	4.9271	0.2201
0.8	84.5216	1.3266	0.0605	4.8	88.8483	5.9761	0.2861
1	86.8982	1.634	0.0965	5.4	89.8233	6.4344	0.3422
1.2	90.067	1.7321	0.1467	6	90.3717	7.8876	0.4108
1.4	90.4936	2.0322	0.156	6.6	90.5545	7.8876	0.4143
1.6	90.5545	2.0322	0.1601	7.2	90.2498	7.8876	0.4176
1.8	90.9202	2.2091	0.1676	7.8	90.6155	9.1642	0.4428
2	90.9811	2.2738	0.171	8.4	90.9811	9.6469	0.4624
2.2	91.042	2.2738	0.1864	9	91.103	9.6469	0.4748
2.4	91.2249	2.0616	0.1992	9.6	91.1639	9.6469	0.4818
2.6	91.4077	2.2804	0.2201	10.2	91.3467	10.1164	0.488
2.8	91.4077	2.2804	0.2247	10.8	91.3467	10.1164	0.488
3	91.4686	2.2804	0.2268	11.4	91.2249	11.4737	0.5034
3.2	91.5905	2.2804	0.2322	12	91.2249	11.4737	0.5169
3.4	91.6514	2.2804	0.2365	12.6	91.3467	11.4737	0.5276
3.6	91.7124	2.2804	0.2505	13.2	91.3467	11.4737	0.5347
3.8	91.7733	2.7221	0.2611	13.8	91.4686	11.4737	0.5606
4	91.8342	2.7221	0.2659	14.4	91.4686	11.4737	0.5607
4.2	91.8952	2.7221	0.2702	15	91.4686	11.4737	0.5733
4.4	91.8952	2.7221	0.2702	15.6	91.4686	13.1955	0.5847
4.6	91.8952	2.7221	0.2897	16.2	91.4686	13.1955	0.5917
4.8	91.8952	2.7221	0.2924	16.8	91.5296	13.1955	0.6032
5	91.9561	2.7221	0.297	17.4	91.5905	13.1955	0.6538



(a) Based on the servos' positions.



(b) Based on the grip's coordinates.

Fig. 4: Analysis of fidelity metrics based on MAD the scenario Simple Moves.

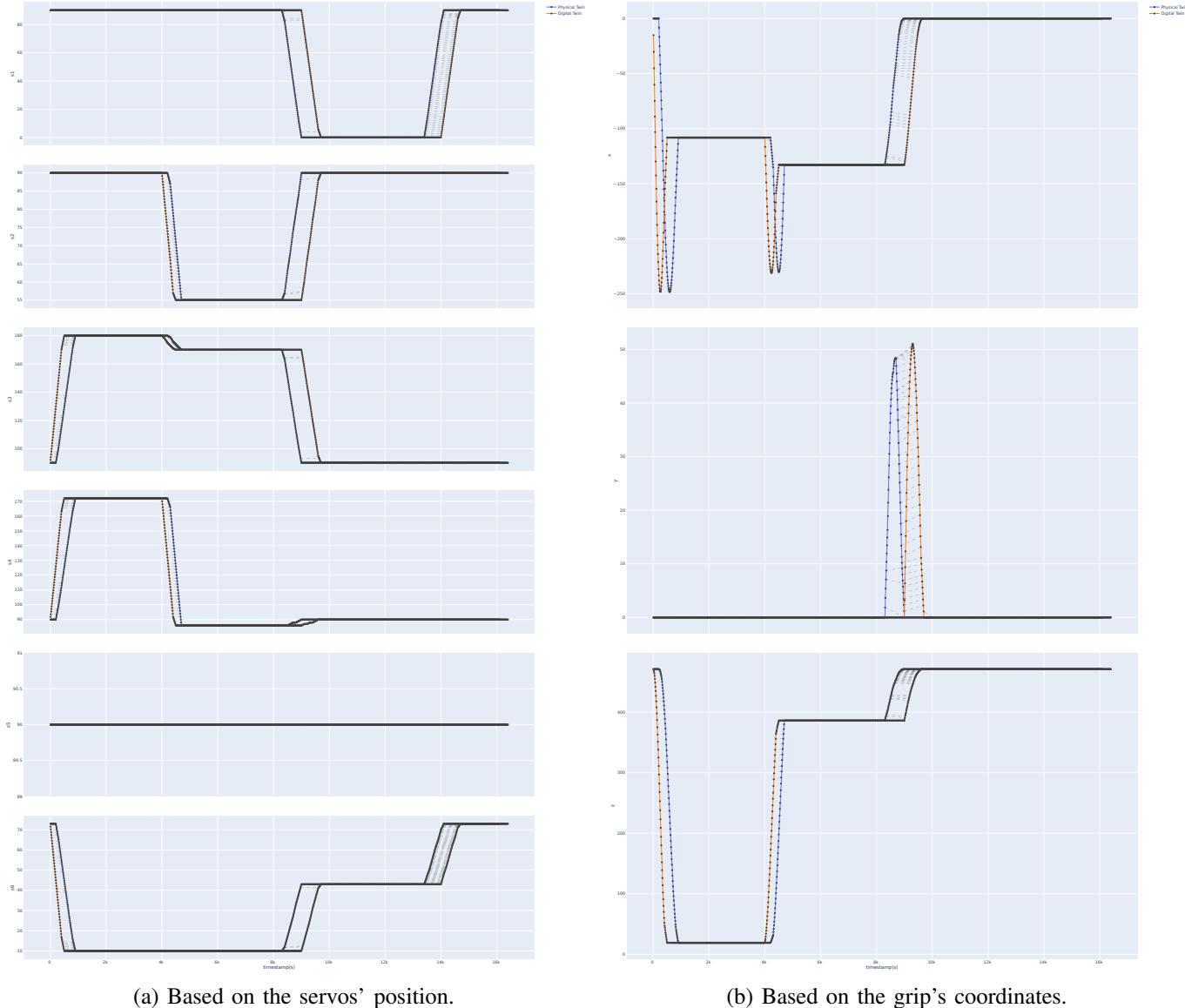


Fig. 5: Alignment of the scenario Simple Moves with MAD of 0.8 degrees and 3 mm, respectively.

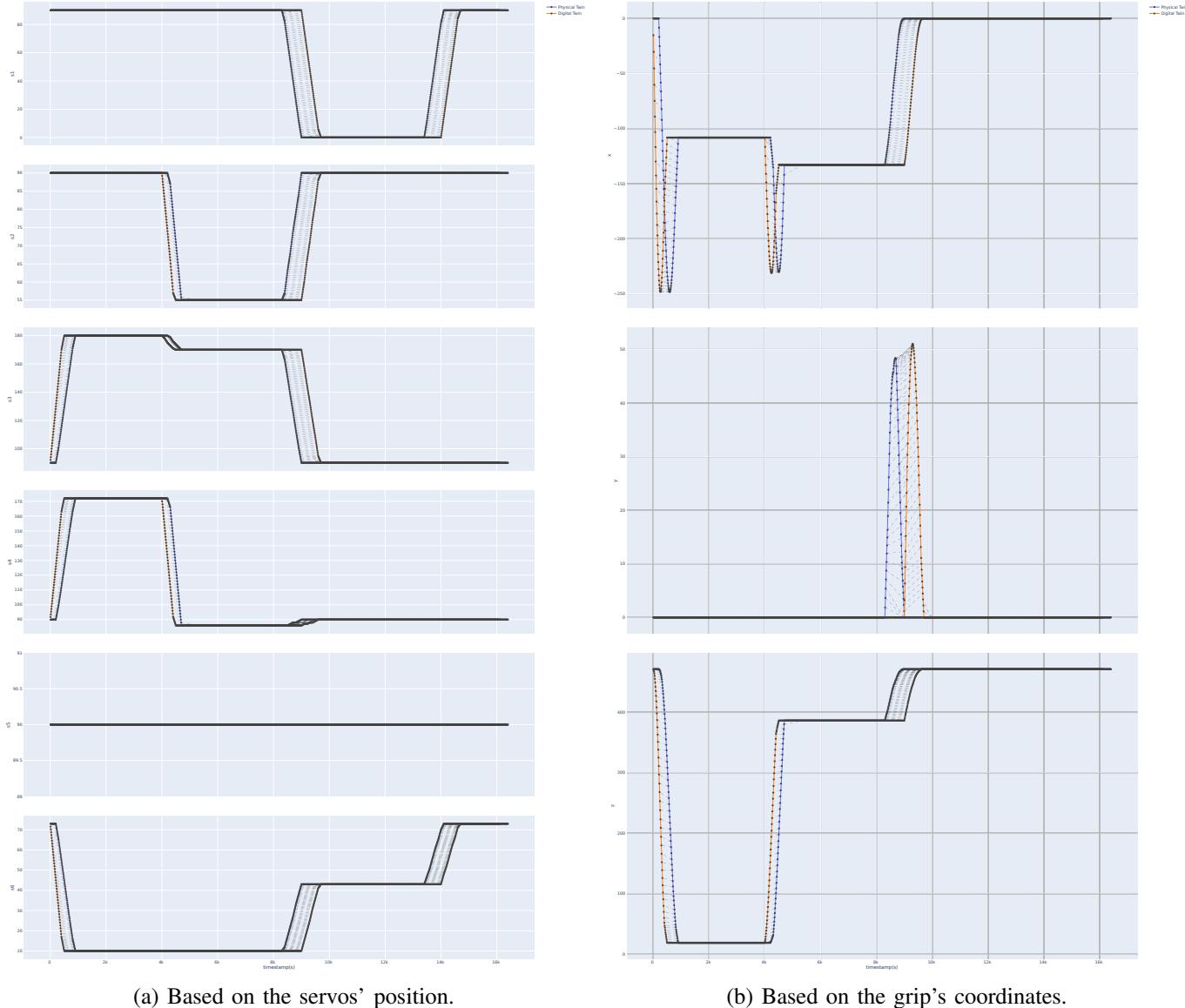


Fig. 6: Alignment of the scenario Simple Move with MAD of 2.6 degrees and 12.5 mm, respectively.

2) Scenario Pick and drop

The results for the three fidelity metrics for different MAD values are available in Table III and a visual representation of this data in Figure 7. The three metrics are available for both representations of the robot's movement: the angle of the six servo motors and the coordinates (x, y, z) of the arm's gripper. In both figures, we have the statistics arranged in ascending order by the value of the MAD.

As we can see, the trend of the metrics for the two representations is similar. It starts with a slow increase in the percentage of aligned snapshots until it reaches a plateau of approximately 78%. This plateau is reached at around 2 and 3 degrees in the case of the servo motor angles, which corresponds to two and three times its precision (one degree). If we analyze the plateau for the coordinates, we also find it between 10 and 15 mm, which is 2 and 3 times the precision (5 mm).

If we look at the distances, we can already observe some signs of low-quality alignment. First, although the value of the Fréchet distance reaches a certain plateau, the Euclidean distance continues to increase with values that double the precision in the case of servo motor angles (2 degrees) or equal its value in the case of coordinates (5 mm). This means that as more snapshots are added, the distances become worse because increasingly dissimilar pairs are incorporated.

To analyze the alignment results more thoroughly, we will examine two of them at different levels of MAD. First, in Figure 5, we will take a MAD level of 0.8 degrees and 3 mm, which yields 85% of aligned snapshots. Upon closer inspection of the figure, we observe a series of aligned points and certain gaps in the curves. It is essential to consider that for a snapshot to be considered aligned, it must have a value within the assigned MAD range for all the attributes.

If we start with Figure 8a, we find that the robotic arm performs transitions faster than our engineering model, as evidenced by transitions with steeper slopes. Similarly, the stopping times are longer for the real system, which is not able to react as quickly as the model does. This leads to intermittent gaps in all movement transitions, where they coincide at some points but not in the entire trajectory due to the difference in speed.

These differences worsen and become more evident in Figure 8b, where, for example, at timestamp 5000, we can see a series of gaps that indicate the difference in slope in the movement along the y-axis, which is performed at different speeds.

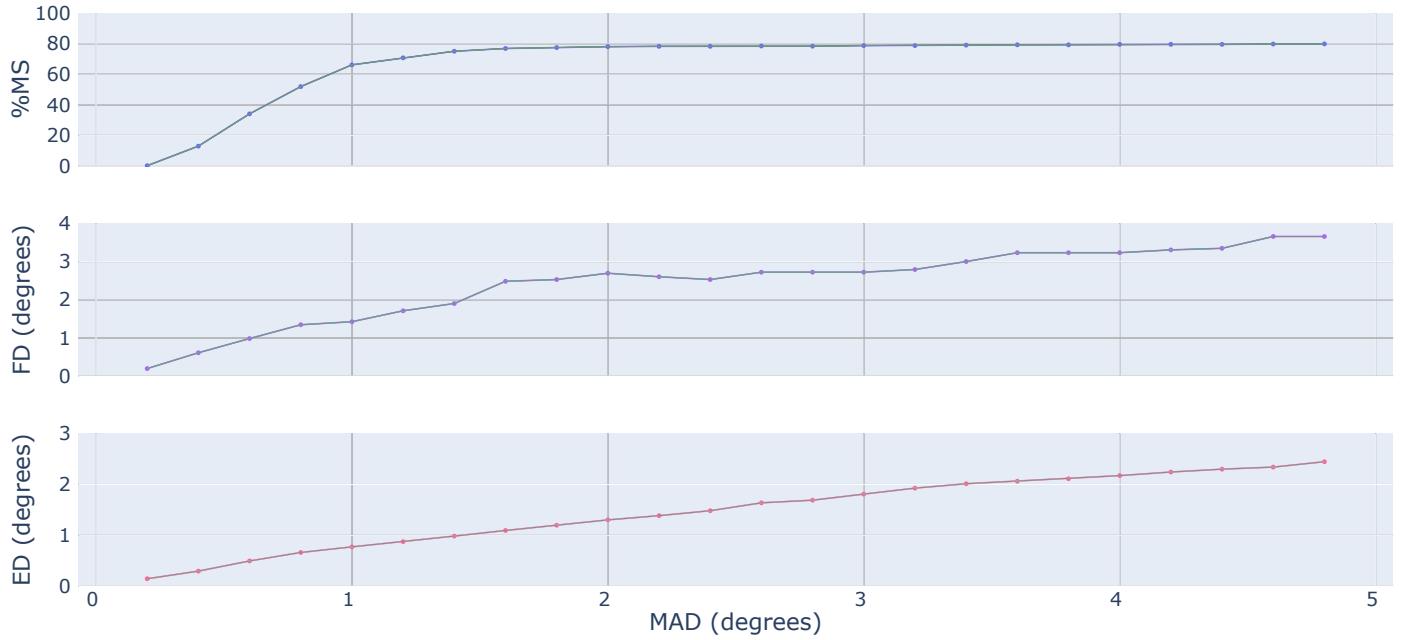
If we increase the MAD level, as seen in Figure 9, we find that these gaps we mentioned earlier are reduced. However, we still encounter intermittent gaps in the slopes due to differences in velocity and the varying waiting times between movements.

With this information, we can draw two conclusions:

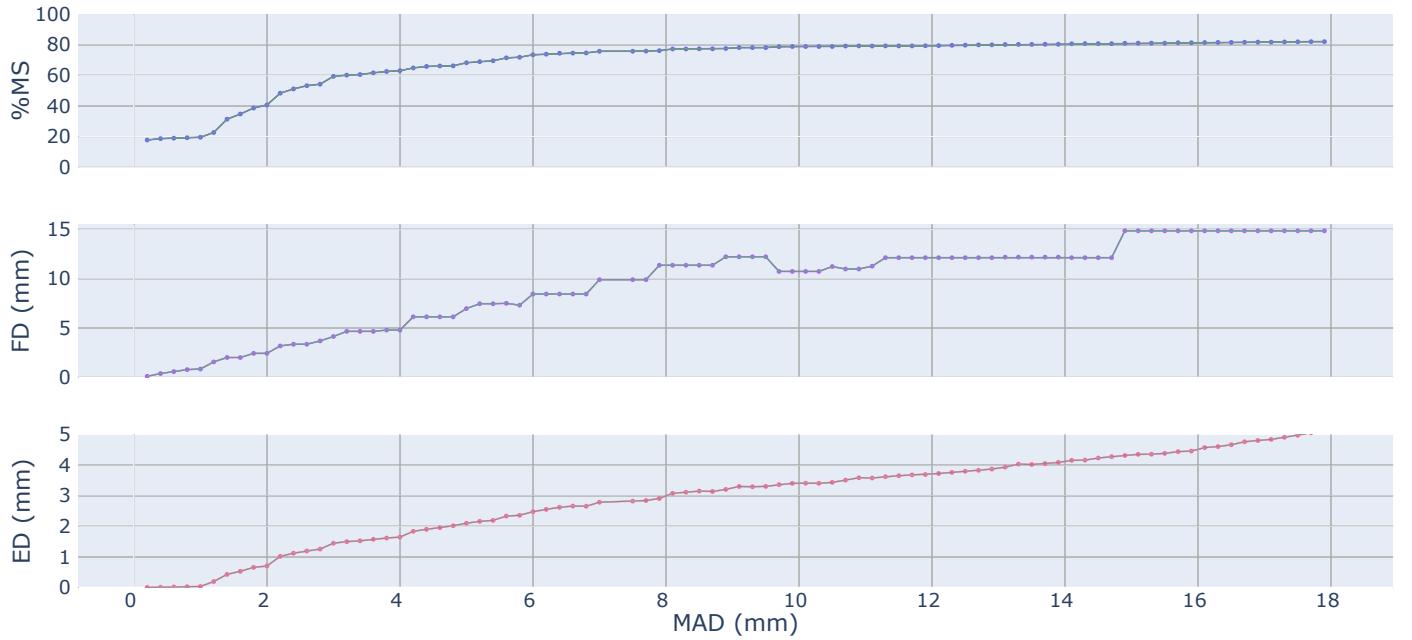
- The plateau in the metrics is reached at a **adequate MAD value that is approximately 2 or 3 times the accuracy of both measuring instruments**. Between 10 and 15 mm for the grip's coordinates and 2 or 3 degrees for the servo's axis angles. Although as a low-quality alignment, we can only observe a plateau in the percentage of aligned points.
- **The percentage of aligned snapshots is below 90 % ($\pm 2\%$)**, which leads us to conclude that the Braccio is not faithful enough to our engineering models in more complex scenarios such as this one.

TABLE III: Fidelity results for scenario *Pick&Drop*.

Axis position (degrees)				Grip's Coordinates (mm)			
MAD	%MS	FD	ED	MAD	%MS	FD	ED
0.2	0.2002	0.2	0.15	3	59.3594	4.1379	1.4488
0.4	13.013	0.6136	0.2998	3.6	61.6617	4.6592	1.5761
0.6	34.1341	0.9849	0.4993	4.2	64.8649	6.132	1.8377
0.8	51.952	1.346	0.6657	4.8	66.2663	6.132	2.0195
1	66.1662	1.4256	0.7758	5.4	69.5696	7.4581	2.1951
1.2	70.7708	1.7114	0.8781	6	73.4735	8.4414	2.4791
1.4	75.0751	1.9018	0.9874	6.6	74.5746	8.4414	2.6629
1.6	76.8769	2.4835	1.0972	7.2	75.7758	9.8982	2.7884
1.8	77.4775	2.529	1.2009	7.8	76.1762	11.3611	2.9115
2	78.0781	2.6943	1.3052	8.4	77.3774	11.3611	3.1524
2.2	78.2783	2.602	1.3883	9	78.1782	12.2197	3.3047
2.4	78.2783	2.529	1.485	9.6	78.7788	10.7362	3.364
2.6	78.4785	2.7217	1.6398	10.2	78.8789	10.7362	3.4086
2.8	78.3784	2.7217	1.6916	10.8	79.2793	10.9802	3.5889
3	78.7788	2.7217	1.8111	11.4	79.2793	12.1281	3.6552
3.2	78.7788	2.7916	1.9282	12	79.4795	12.1281	3.7279
3.4	79.0791	3	2.0146	12.6	79.98	12.1281	3.8339
3.6	79.2793	3.2311	2.0641	13.2	80.2803	12.1824	4.0356
3.8	79.2793	3.2311	2.1129	13.8	80.3804	12.1824	4.0914
4	79.5796	3.2311	2.1741	14.4	80.7808	12.1281	4.2357
4.2	79.5796	3.305	2.2446	15	81.0811	14.8537	4.356
4.4	79.5796	3.3461	2.3005	15.6	81.3814	14.8537	4.4424
4.6	79.7798	3.6556	2.3414	16.2	81.5816	14.8537	4.6086
4.8	79.8799	3.6556	2.4465	16.8	81.7818	14.8537	4.808
5	79.98	3.6556	2.4533	17.4	81.982	14.8537	4.9791



(a) Based on the servos' positions.



(b) Based on the grip's coordinates.

Fig. 7: Analysis of fidelity metrics based on MAD the scenario Pick&Drop.

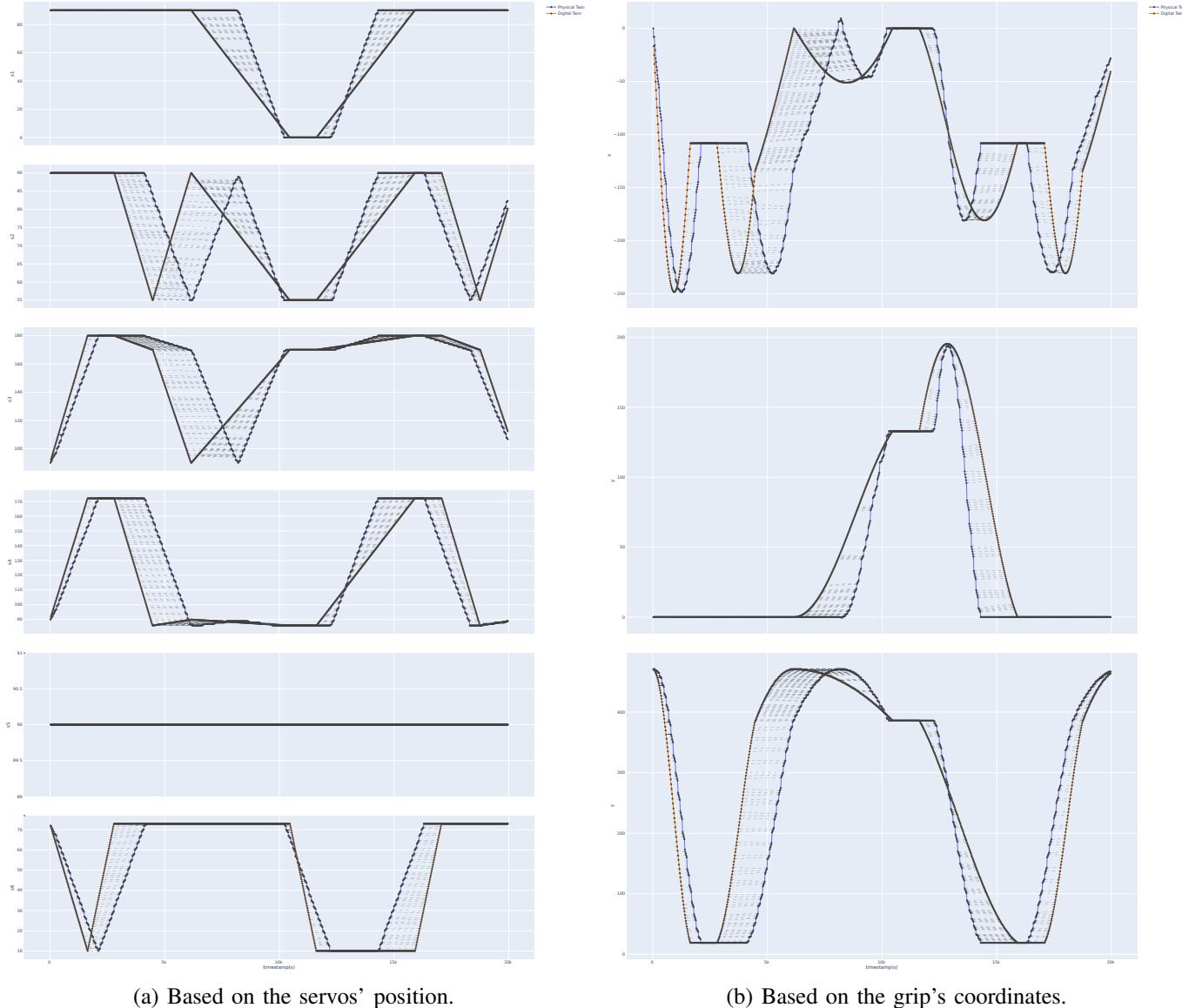
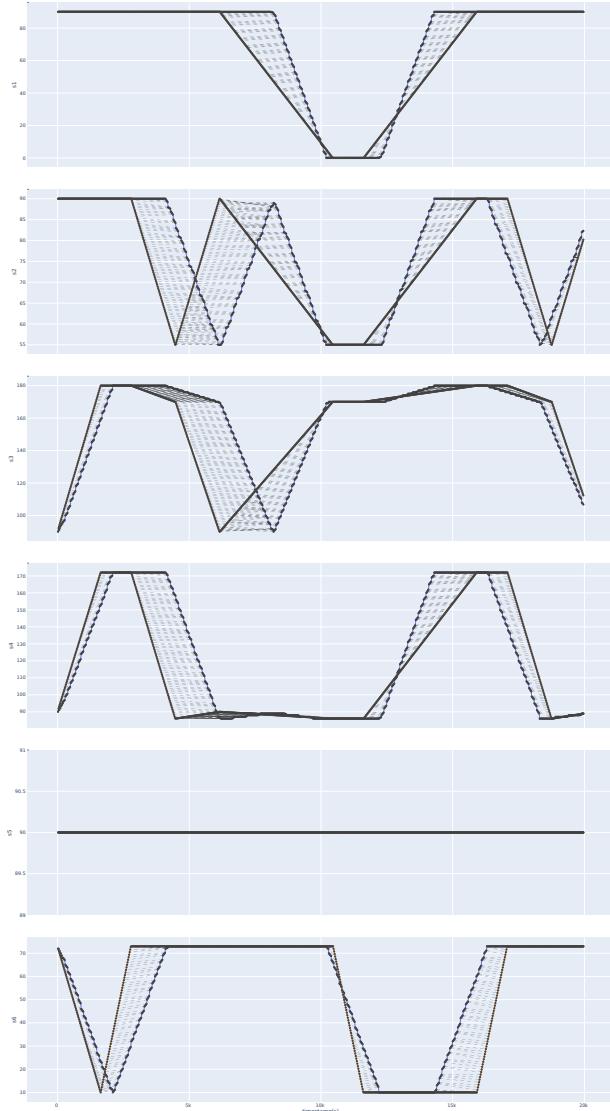
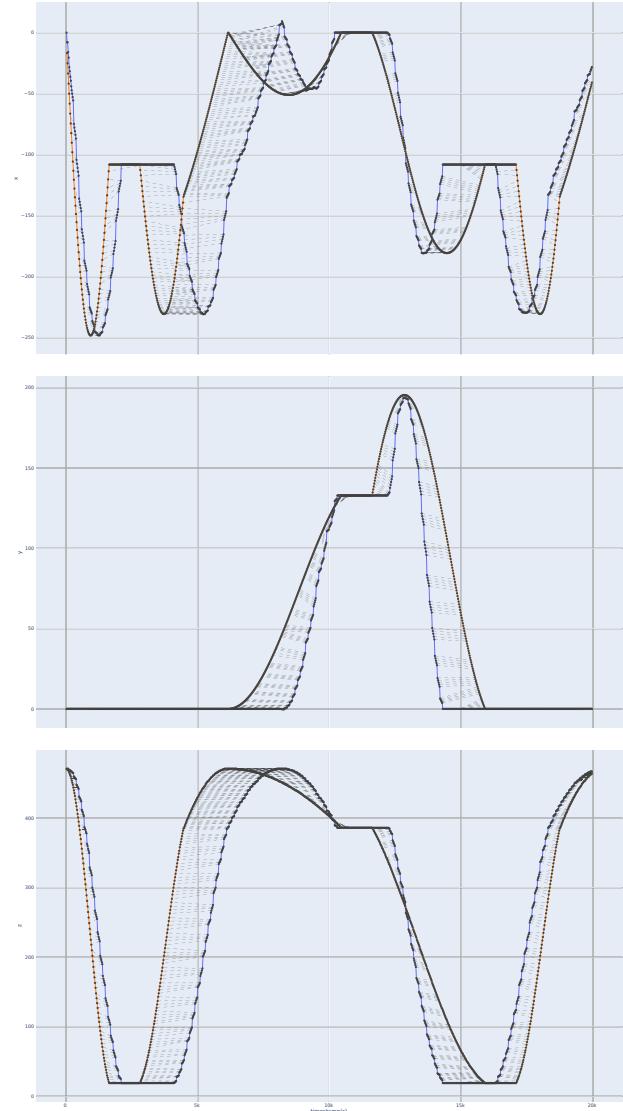


Fig. 8: Alignment of the scenario *Pick&Drop* with MAD of 0.8 degrees and 3 mm, respectively.



(a) Based on the servos' position.



(b) Based on the grip's coordinates.

Fig. 9: Alignment of the scenario *Pick&Drop* with MAD of 2.6 degrees and 12.5 mm, respectively.

References

- [1] P. Muñoz, J. Troya, and A. Vallecillo, “Using UML and OCL Models to Realize High-Level Digital Twins,” in *Proc. of Mod-DiT2021@MODELS’21*. IEEE, 2021, pp. 212–220.
- [2] D. Pérez-Porras, P. Muñoz, J. Troya, and A. Vallecillo, “Key-Value vs Graph-based data lakes for realizing Digital Twin systems,” in *Proc. of MeSS@STAF’22*, 2022.
- [3] M. Gogolla, F. Büttner, and M. Richters, “USE: A UML-based specification environment for validating UML and OCL,” *Sci. Comput. Program.*, vol. 69, pp. 27–34, 2007.
- [4] “Example digital twin: Arduino braccio,” 2022. [Online]. Available: <https://github.com/daikoderu/uma-braccio-digital-twin/>
- [5] P. Muñoz, J. Troya, M. Wimmer, and A. Vallecillo, “Using trace alignments for measuring the fidelity of a physical and a digital twin: General concepts,” 2023. [Online]. Available: https://github.com/atenearesearchgroup/fidelity-measure-for-dts/blob/main/docs/Technical_Report_General_Concepts.pdf
- [6] I. Korf, M. Yandell, and J. A. Bedell, *BLAST - an essential guide to the basic local alignment search tool*. O'Reilly, 2003.
- [7] D. A. Snow, Ed., *Plant Engineer's Reference Book*, 2nd ed. Elsevier, 2003.