Sensor fusion on self-reconfigurable modular robots

ROBOTICS SEMESTER PROJECT I

RECONFIGURABLE ROBOTICS LAB

PROFESSOR: JAMIE PAIK
SUPERVISOR: KEVIN HOLDCROFT
STUDENT: CHRISTOPHER J. STOCKER

December 4, 2024

Abstract

Self-reconfigurable modular robots hold great potential for the future of versatile and adaptable systems. However, new modular systems add new mechanical and electronic complexities to traditional robotic systems. Single unit errors are compounded by the increasing number of modules, and without fail-safes these can be propagated throughout the entire system. Nevertheless, modular systems have built-in redundancies that increase with each added unit. In order to improve the mechanical and software robustness, these systems could leverage their redundancies to mitigate errors and improve system's state estimation. The ideal working scenario is one where a multi-modular system has better and more accurate picture of the robot's state than any singular unit alone could.

This paper analyzes the wrong sensor readings from the absolute magnetic encoder in the Mori 3, which are triggered when an external torque is applied between two connected modules. The mechanical compliance between the connecting edges create an axis misalignment around the encoder that produce faulty sensor readings. These error are provoked tested and quantified. A simple yet elegant model for absolute angle detection was developed using the Mori's accelerometer (IMU) sensor readings to try and solve this issue. This required some considerable testing, and refinement to validate and improve the model's results. Lastly, a video motion capture tool (*Kinovea*) was used to benchmark and validate the final model's results.

Acknowledgments

I would first like to express my sincere gratitude to Prof. Jamie Paik for allowing me to pursue this project in her lab. Her continuous involvement and support throughout the semester in the biweekly Monday group meetings were comparable to no other professor I have known. Thanks for bearing with us all throughout the semester.

My work could not have been made possible without the help and support from my project supervisor Kevin Holdcroft. His generous time, feedback, and patience to make new measurements when I needed them were invaluable to the realization of this project. His good humor and witty puns always brought liveliness to the room.

Similarly, I would like to thank postdoctoral research assistant Anastasia Bolotnikova. For her kindness to not only share the Mori lab-room but also for her observations and recommendations.

Lastly, to my robotic misfits this semester Alfred, Bob, Carl, Darlene, Elaine, Gina, Jerry and Hiro, you bots were always plotting something.

1 Introduction

Background

Imagine rebuilding a Formula One car not in a pit lane, but as you speed across the race track. That is the challenge edge computing rises. We firmly live in the age of cloud computing. We use centralized services such as Netflix, Apple TV, Gmail, Slack, etc. Edge computing is all about bring the computation closer to the source of data instead on of relying on one centralized computer to crunch all the numbers and send the results back. An everyday example of this is is Amazon's speech recognition Alexa, who sends compressed representation of your processed speech to the cloud where this decompresses it, analyses it, and sends back a response back. A specialized chip could process more of the data locally on the device instead of relying entirely on the cloud's computational power. This is why, the next generation of smart devices is increasing looking to processing data locally on the hardware. This liberates bandwidth space and lowers latency between devices, thus enabling larger scalability and efficiency.

Problem Statement

The ultimate example of edge computing are fully autonomous systems such as self driving cars, drones, and robots. Autonomous robotic systems typically carry numerous sensors. Such systems can not rely on sending all their information to one central computer and wait for it to return an answer. This problematic becomes even more complex when considering modular robotic systems as an answer has to be transmitted to all relevant modules. The solution is to handle most of the data processing on the device and to transmit less but more relevant data.

For my semester project I was tasked with improving the sensor readings from the Mori's angular position sensor. The Mori is a modular reconfigurable robot (MRR). MRR are made up from many repeating modules that can be rearranged together to form new configurations. MRR have three primary areas of promise: versatility, robustness, and low cost. Up to date MRR have only been able to successfully deliver on one of these promises; versatility. Thanks to its redundant degrees of freedom MRR can easily adapt their configuration to their environment [1].

Most robotic systems today use an integrated design approach as these systems can be optimized for higher performance than those with a modular design. This raises into question the second promise of modular robots: their robustness. The redundancy and self-configuration of these system should be leveraged to fix both mechanical and system errors. A faulty module could easily be swapped with a new one, and an odd sensor reading could immediately be cross validated with another module.

One of the challenges modular systems have is error multiplication. Without error verification controls and redundancy loops, the error of one module could become the error of many. Small sensor variations can be compounded and propagated throughout the entire system. These errors are sometimes difficult to detect as they occasionally require multiple systems to be connected. These errors can originate from small fabrication intolerance. For example, a small sensor offset in the IMU may cause sensor drift while the system remains in invariant conditions. Also mechanical

intolerances in the coupling alignment or module's elasticity could provoke wrong sensor readings.

Figure 1: Depiction of error propagation through connected Mori modules.
(Image courtesy of K. Holdcroft)

It is important to understand that these systems also have to be able to handle module failure, as the number of modules increases the chance of module failures and communication errors increase. In fact, modules will undoubtedly fail given a large quantity of them. Assume for example the probability that a single module fails is p_1, the probability that one module out of n modules from failing is calculated by equation below.

$$p_n = 1 - (1 - p_1)^n$$

If we take ten modules (n=10) with a probability of a single unit of failing as 1%, then the probability of one out of the n failing is p_n, is equal to 9.6%. Given this context, we come to realize that inter-modular redundancy loops and fail-safes are absolutely necessary [2].

My work on this project was two part. First, I needed to understand why the angle sensor reading on the Mori deviated from its true value. Second, I needed to find a way to improve the Mori's angle sensor by using other available sensors on Mori and developing its corresponding software.The end goal is to increase the Mori's robustness by incorporating inter-modular communication.

Importance

Figure 2: System robustness through inter-modular sensor sharing
(Image courtesy of K. Holdcroft)

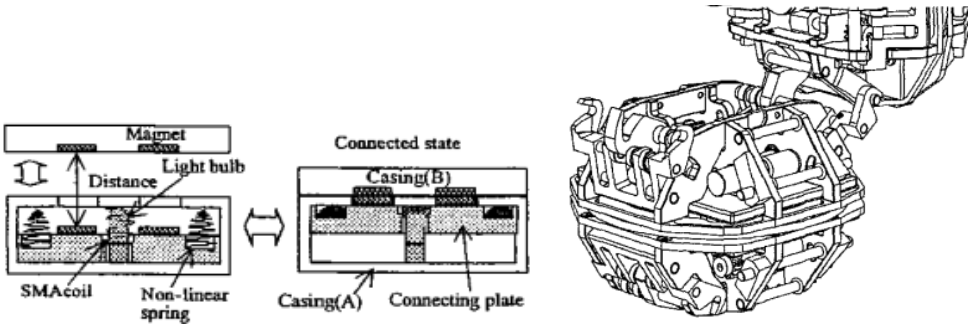
This work is key because without accurate sensor measurements a precise control of the Mori can not be guaranteed. MRR still have to fulfill their second promise of robustness. As previously shown MRR have a larger probability of unit failure with increasing number of units. This is also true for Mori robots. However, hope lies in taking advantage of inter-modular redundancies to mitigate these failures. More modules should ultimately improve the system not make it worse. More modules add more sensors to the system increasing its redundancy. Sensor and data fusion has a key role to play to improve state estimation and localization. Quick inter-modular communication and localized computation will create more reliant and robust systems. This could accelerate the promise of one day creating not only versatile but also robust MRRs.

State of the Art

In order to place the Mori robot within the scientific literature, we focus our review to robotic systems implementing reconfigurable structures and their connecting mechanisms. The practical effectiveness of modular robotic systems rely strongly on the connection mechanisms used to join the separate modules. The majority of controllable connection mechanisms can be categorized as mechanical, magneto-mechanical, or electro-magnetic. These mechanisms can further be divided into gendered, bi-gendered or genderless connections [3].

Gendered Connections

These connections have two distinct complementary surfaces that puzzle together. One surface has an active element, such as a latch or electro-magnet, and the complementary surface contains passive elements, such as loops or permanent magnets. Examples of this type of connectors are found in older robotic system such as the M-Tran [4] and ATRON robots [5]. Gendered connections constrain the connecting surfaces to be complementary, thus bi-gendered and genderless connectors have been adopted to improve modulatority.



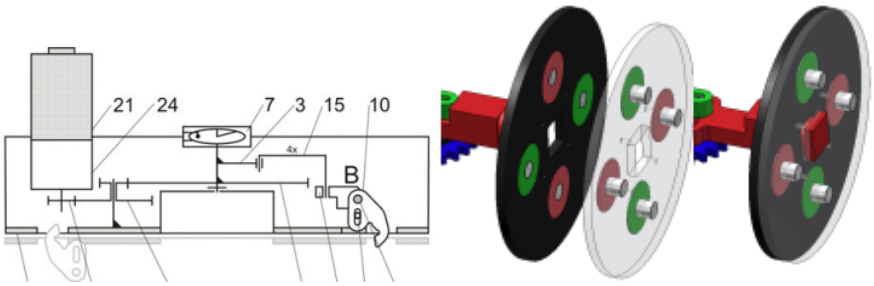
((a)) M-Tran magnet connection mechanism (Image source: [4]).

((b)) ATRON latch connection mechanism (Image source: [5]).

Figure 3: Gendered connection mechanisms

Bi-gendered Connections

This connection mechanism is an extension of the gendered design. It incorporates both active and passive elements into a single connector surface. Bi-gendered mechanisms allow all connectors on a module to be identical, thus removing constrains on which surfaces are able to connect together. Additionally, it only requires one side to actuate in order to make a connection, other latch-based mechanisms require both sides to disconnect in order to break the connection. Examples of this type of connector are found in the Roombots [6] and SMORES [7] robotic systems.



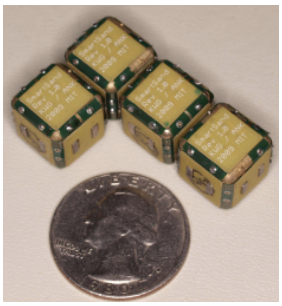
((a)) Roombots mechanical latch connection mechanism (Image source: [6])

((b)) SMORES magnetic-latch docking and coupling mechanism (Image source: [7]).

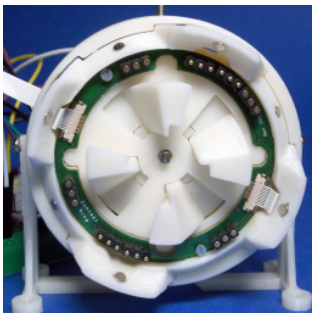
Figure 4: Bi-gendered connection mechanisms.)

Genderless Connections

Genderless mechanisms distinguish themselves by having active elements that mate with their neighboring active elements, such that either surface can disconnect without the action from the other. These types of connections are preferred in order to achieve reconfigurability at each interface with fully modular systems. Examples of this type of connector are found in the Mori [8], HiGen [3], and Pebbles [9] robotic systems.



((a)) Pebbles electropermanent magnet connection mechanism (Image source: [9]).



((b)) HiGen latch connection mechanism (Image source: [3]).

Figure 5: Genderless connection mechanisms

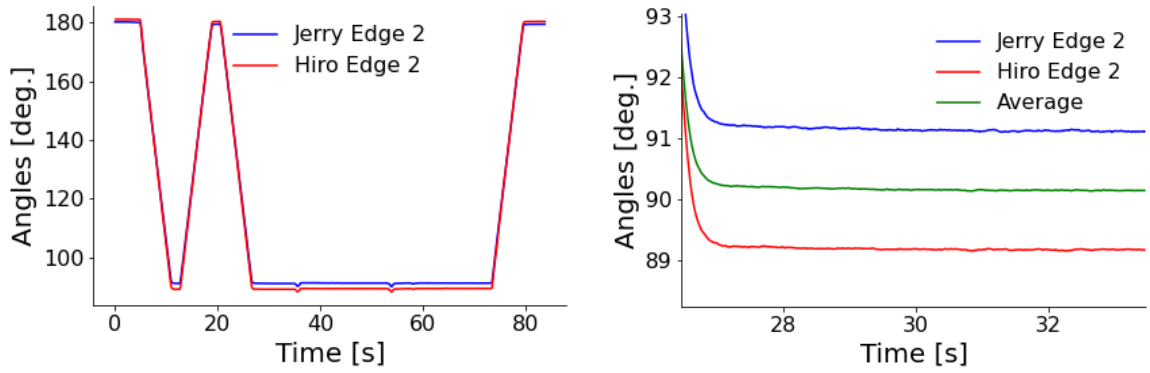
A comparison of connectors of modular robots supporting self-configuration is presented below. As time has passed these systems have transitioned from gendered, to bi-gendered, and now genderless architectures.

System / Connector	Category	Gender	Year	Coupling Time (s)
ATRON [5]	Mechanical	Gendered	2004	2.4
M-TRAN I/II [4]	Mag-Mech.	Gendered	2003	60 to 180
Roombots [6]	Mechanical	Bi-gendered	2009	2
SMORES [7]	Mag-Mech.	Bi-gendered	2012	0.8 to 2.3
Pebbles [9]	E-Magnetic	Genderless	2010	0.0003
HiGen [3]	Mechanical	Genderless	2014	0.2
Mori [8]	Mechanical	Genderless	2017	3

2 Methodology

Software Complexities

The Mori system reads the same absolute angle between two connections using two encoders. Each encoder returns a slightly different angle readings. The Mori system takes the average of these two readings as the true absolute angle between two connections. The control system uses this averaged value for its state estimation. For example, in a given test two modules Jerry (blue) and Hiro (red) were placed flat at 180° and commanded to iterate to 90° and back flat again, as shown in Figure 6(a). Jerry and Hiro returned different values for the absolute angle with Jerry sensing 91.5° and Hiro sensing 89.4°. The average between these two values ends up to roughly 90°, which is what the control system system stabilizes at. Figure 6(b) shows the control system follows the averaged reading as the absolute angle between Jerry and Hiro.



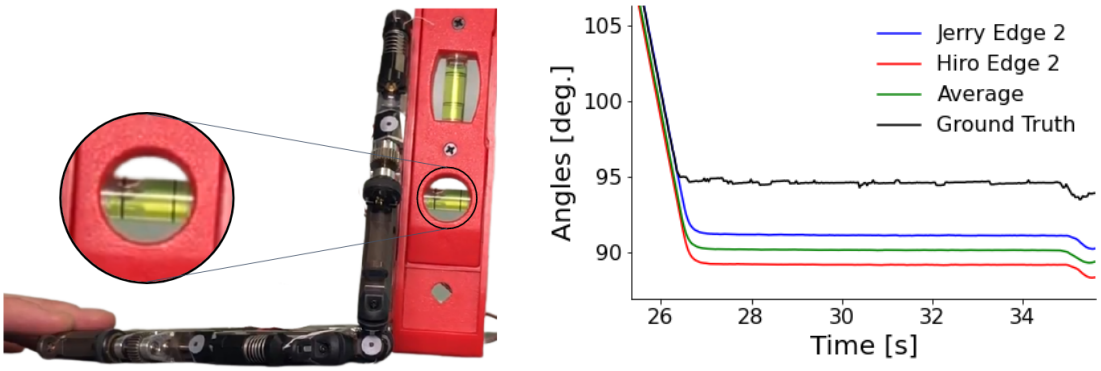
((a)) Jerry and Hiro 90°test.

((b)) Zoom to averaged sensor value of 90°.

Figure 6: Sensor intolerances of the angle encoders.

Hardware Complexities

During experimentation and testing of the returned angle values of the Mori, it became clear that under certain situations the averaged angle the Mori were computing was off from the ground truth. For example during the same experiment of Figure 6 a qualitative 90° angle test was done. As previously mentioned, the Mori were commanded to have a right angle between the time interval of 25s-73s. During this same interval a level was used to verify the angle was actually 90°. As shown in Figure 7(a) the angle between Jerry and Hiro was not 90° despite recording otherwise. To further quantify this error the sensed value was plotted against the ground truth on Figure 7(b)). The results showed that the true angle was larger than 90°. The angle difference between the the Mori's sense value and true value was of about °.



((a)) Qualitative level test for $90°$ measure.

((b)) $5°$ angle difference between sensed and true value.

Figure 7: Right angle test.

These sensor errors were found when the Mori needed to only carry its own weight. Next we found errors provoked when an external torque was added to the connecting hinge axis between two modules. Although this torque and torsion to the system was manually provoked one can imagine the system experiencing similar torque when working as a cantilever supporting the weight of its connected neighbors.

The angle between two connected modules is measured by an absolute magnetic encoder AMS AS5048B. Each edge has a permanent magnet and a magnetic encoder as shown in Figure 8. The encoder detects the magnetic field of the permanent magnet on the opposing module.

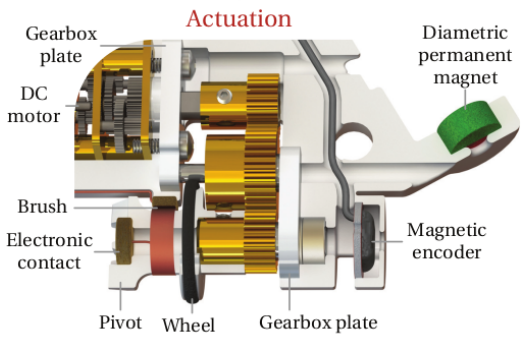


Figure 8: Design of the absolute magnetic encoder on the Mori (Image source: [8]).

.

Absolute magnetic encoders were used, since as soon as two modules connect, each module must be aware of the absolute angle between them. If the sensing source and sensor were on the same module then the sensor could only relay information of the pivot rotation. In the Mori's design the cylindrical, diametric magnet is placed inside the housing, with the same orientation in each module. The absolute magnetic encoder from another module can sense the angle between the two modules once their axes are aligned [8]. This process is shown in Figure 9.

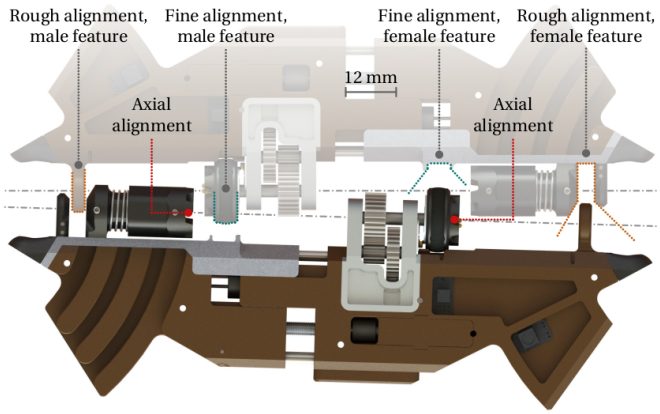
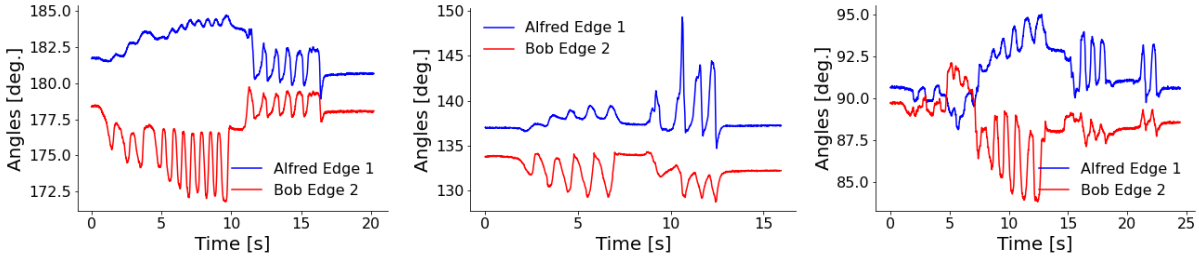


Figure 9: Edge alignment on the Mori (Image source: [8]).

We tested the angle measurements when an external torque was applied between the modules' hinges. Small mechanical elasticity and compliance inside the modules allowed to introduce small misalignment between the modules. The modules would slightly budge and this was enough to off-center the encoder with the permanent magnet. These tests were done on two connected modules Alfred and Bob at three different resting angles (180°,135°,90°).



((a)) Angle error at 180°. ((b)) Angle error at 135°. ((c)) Angle error at 90°.

Figure 10: Forced angle errors at different resting angles.

The plots showed how a slight misalignment between the angle encoder and the permanent magnet could create large angle error readings. It is interesting to note that these error were occasionally symmetric. This suggests that as one sensor got closer to the permanent magnet the opposite sensor moves away.

Theory

A large part of my project focused on obtaining an angle reading from two connected modules using their respective inertial measuring unit (IMU). Each Mori carries a MMA8452Q, 3-axis, 12-bit/8-bit digital accelerometer. The question was how can we get more information from the Moris to obtain better angle readings between two modules. The first step is to understand the readings an individual IMU returns. This took some time diving into the MMA8452Q's data sheet and the Mori's programming. Two observations were made.

Higher Resolution

First observation was that the MMA8452Q can be optimized for lower power modes or for higher resolution of the output data. For our purpose we want a fast and accurate reading of the IMU sensor. For this we needed to change the programming of the **LNOISE** bit in register **0x2A**. This improves the resolution but at the expense of losing dynamic range. Normal dynamic range is of $\pm$ 4g, by activating the high resolution mode, more accurate readings can be made between $\pm$ 2g. This change also affects all internal functions of accelerometer to reduce sensor noise [10].

Oversampling

The second observation made was that the resolution of the data could be improved by oversampling. There are four different sampling modes in the MMA8452Q. The four different modes offer different options for compromising between resolution and current consumption. For our purpose the current consumption was ignored and the mode with highest resolution was chosen. This option is achieved by changing the modes bits to (**MODS = 10**) in the register **0x2B**. The highest resolution is achieved at a sampling rate of **1.56 Hz** [10]. The trade-off for higher resolution at the expense of more power consumption should be considered when the Mori are running on batteries alone.

The next step is to understand what readings the IMU makes. The IMU returns signed values between $\pm$ 2048, which are then shifted to unsigned integer values between [0-4096]. This shift was previously done in order to maximize computational power. These values (*counts*) are then transformed to accelerations values through a linear transformation shown below on eq(1).

$$acceleration_{x,y,z} = \frac{counts}{1024} - 2g \tag{1}$$

Now that we retrieved the acceleration values in x, y, z it is important to understand how the accelerometer is positioned on the Mori as this sets in place a corresponding reference frame. Figure 11(a) shows where the accelerometer is placed on the Mori and Figure 11(b) illustrates the corresponding reference frame. For simplification purposes the reference frame's origin was translated to the triangle's centroid.

((a)) Accelerometer placement ((b)) Reference frame

Figure 11: Small IMU deviations

The next task was to determine how to extract meaningful angle values from the acceleration readings. Accelerometers are commonly used to calculate a tilt angle or the angle needed to determine the orientation of an object (ex. phone orientation). It is important to note that the output from accelerometers is not linear but follows a sine-wave. One cannot take the direct output as a proportional representation of an angle of tilt based on gravity. This relationship is illustrated on Figure 12. When the axis points in the same direction as gravity it's corresponding acceleration value is equal to 1g. When the axis points perpendicular to gravity the corresponding acceleration value is equal to 0g.

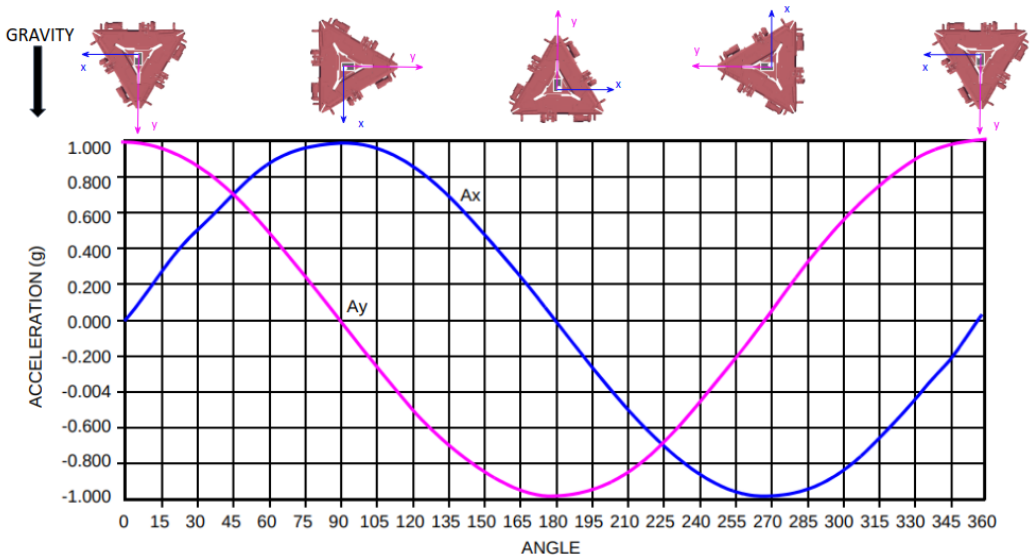


Figure 12: Acceleration [g] to angle degree relationship.

Although, angle tilt could provide an alternate method for finding the absolute angle between two modules, the resulting model would need to keep track of three angles for the three axis of rotation (pitch, roll and yaw) for each module. To calculate the absolute angle, a change of bases through matrix calculation would need to be applied. Once the modules move in 3D space careful calculation of the corresponding tilt angle in x, y, and z, from both modules would be needed. Although feasible a more elegant solution was found by taking a different approach to the accelerometer readings. Consider the Cartesian equation for a plane α as shown by the equation 2:

$$ax + by + cz + d = 0, \quad a, b, c \in \mathbb{R} \tag{2}$$

The normal vector of the plane α containing the point A is defined by the equation 3. By definition this is a vector that is perpendicular to any line $\overrightarrow{AM}$ on that plane:

$$M \in \alpha \iff \overrightarrow{AM} \cdot \vec{n} = 0 \tag{3}$$

The normal vector is defined as the coefficients of the Cartesian plane previously defined on equation 2.

$$\vec{n} = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \tag{4}$$

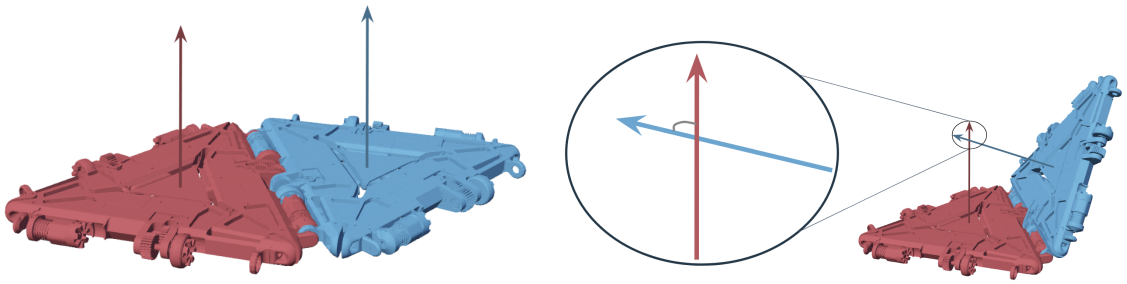
Moreover, the angle φ between two planes α and β is also the angle between the normal vectors $\overrightarrow{n_\alpha}$ and $\overrightarrow{n_\beta}$.

$$\varphi = \angle(\alpha, \beta) = \angle\left(\overrightarrow{n_\alpha}, \overrightarrow{n_\beta}\right)$$

hence, φ can be determined by using equation 5:

$$\cos\varphi = \frac{\overrightarrow{n_\alpha} \cdot \overrightarrow{n_\beta}}{\left\|\overrightarrow{n_\alpha}\right\| \left\|\overrightarrow{n_\beta}\right\|} \quad 0 \leq \varphi < \pi \tag{5}$$

Now for our robotic system consider each module's orientation described by a plane and its corresponding normal vector. Suppose $\overrightarrow{n_\alpha}$ describes the normal vector of the plane containing the red Mori and $\overrightarrow{n_\beta}$ describing the plane containing the blue Mori as shown in Figure 13.



((a)) Normal vectors $\overrightarrow{n_\alpha}$ and $\overrightarrow{n_\beta}$.

((b)) Angle φ between $\overrightarrow{n_\alpha}$ and $\overrightarrow{n_\beta}$.

Figure 13: Normal vector model

I used the acceleration components in x, y and, z to correspond to a planes normal vector components, as shown in equation 6.

$$\overrightarrow{n_\alpha} = \begin{pmatrix} \ddot{x}_\alpha \\ \ddot{y}_\alpha \\ \ddot{z}_\alpha \end{pmatrix}, \qquad \overrightarrow{n_\beta} = \begin{pmatrix} \ddot{x}_\beta \\ \ddot{y}_\beta \\ \ddot{z}_\beta \end{pmatrix} \tag{6}$$

We can then calculate the absolute angle value φ between two modules by using 7.

$$\cos\varphi = \frac{\ddot{x}_\alpha \cdot \ddot{x}_\beta + \ddot{y}_\alpha \cdot \ddot{y}_\beta + \ddot{z}_\alpha \cdot \ddot{z}_\beta}{\sqrt{\ddot{x}_\alpha^2 + \ddot{y}_\alpha^2 + \ddot{z}_\alpha^2}\ \sqrt{\ddot{x}_\beta^2 + \ddot{y}_\beta^2 + \ddot{z}_\beta^2}} \tag{7}$$

Implementation

To better understand how the system works consider the following example. Imagine two connected Mori modules laying flat with their corresponding reference frames as show in Figure14.

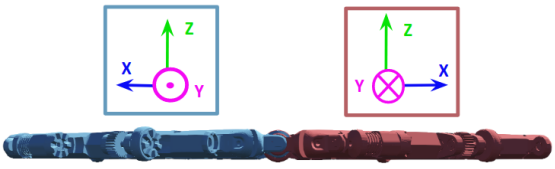


Figure 14: Initial reference frames at 180°.

Using the model described before, their corresponding normal vectors are

$$\overrightarrow{n_{blue}} = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}, \qquad \overrightarrow{n_{red}} = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$$

Next imagine the blue Mori performing a 90° degree transformation as show in Figure 15. The gravity vector will have switched from the Z-axis to the X-axis.

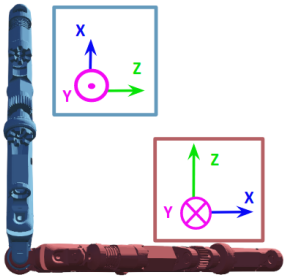


Figure 15: Reference frame transformation at 90°.

If we go back to the original frame this change would have created corresponding normal vectors that are perpendicular to each other.

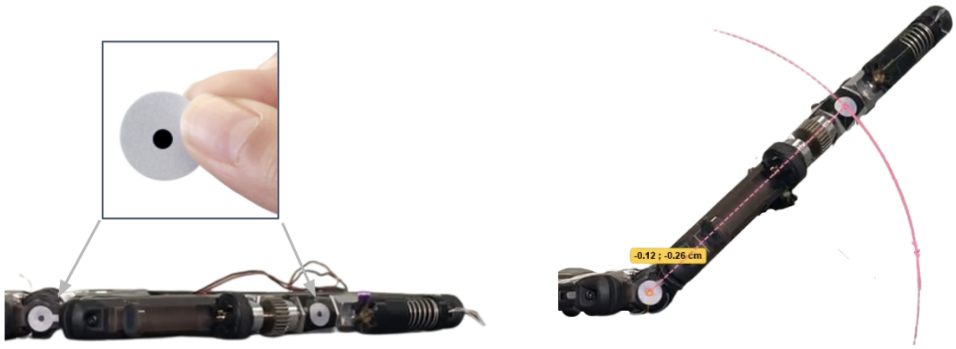
$$\overrightarrow{n_{blue}} = \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}, \qquad \overrightarrow{n_{red}} = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$$

The corresponding angle between these two normal vectors correctly evaluate to 90°.

$$cos\varphi = \frac{0}{\sqrt{1}\ \sqrt{1}} \implies \varphi = 90^{\circ} \tag{8}$$

3 Results

Experimental Setup



((a)) Motion tracking reflective markers.

((b)) Motion capture demonstration on Kinovea.

Figure 16: Motion Capture experimental setup.

To verify the results from the IMU model we had to find a way to obtain the true absolute angle between two modules. In other words we needed a way to retrieve the ground truth. For this I used a free and open source software called *Kinovea*. This program features utilities to capture, slow down, annotate, and measure motion in videos. This programmed was new to me so finding the right markers to capture the motion was an ongoing learning process. Since the angle difference between the magnetic encoder and IMU model was at times only off by a couple degrees, we needed a ground truth of high fidelity to discern which reading was correct.

This meant spending at times more effort on the right positioning of the camera and modules than the actual experiment itself. However this was crucial as learning from experience a slight variation of the camera angle from the horizon skewed all the results. The motion capture software required meticulous attention to position the camera such that the plane of motion was perfectly parallel to the film frame.

Initially to track the rotation of the module, specific frame parts of the Mori were used as markers (ex. the gearbox) for the program to track. The program would then create a CSV file of the angle between said markers. However, once the modules started to move, changes in lighting and orientation on these markers would cause the program to lose track of them. The program would drop the markers all together or randomly select a different point to track.

In order to solve this problem the angle tracking functionality of the program was swapped by the position tracking utility. Then new markers were added to mitigate the changes of light and orientation. The solution was to use markers that appeared the same under rotation (i.e. circular sticker markers). These were placed on the hinge axis and on an adjacent side as show in Figure 16(a). To add pin point precision to our tracking, a black dot was drawn on the center of the circular markers and this point was marked for tracking. With these changes in place the position detection of the markers worked reliably well as shown in Figure 16(b). Using simple trigonometric calculations the absolute angle was retrieved by using the location of the markers.

Results

The results from the experimental setup described before are plotted on the Figure 17. This graph plots the different angle reading between the IMU angle model and the ground truth retrieved from the motion capture software.

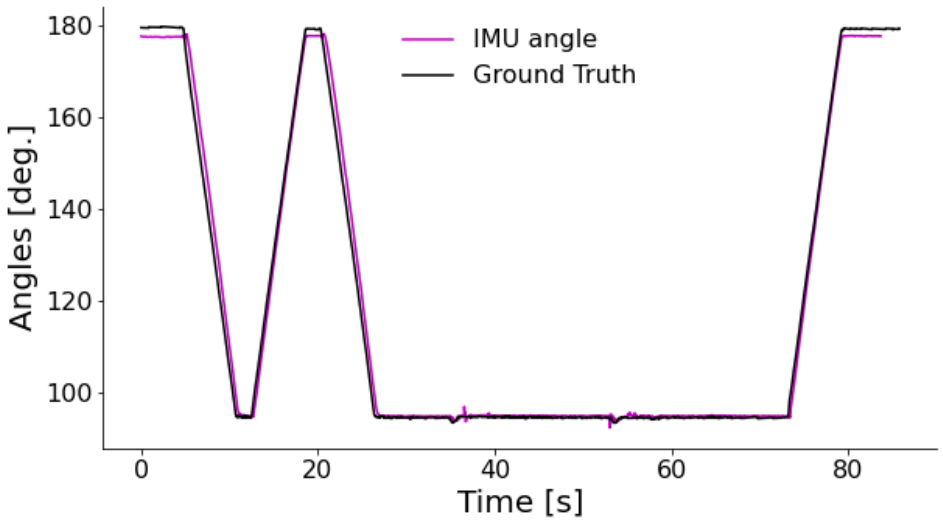
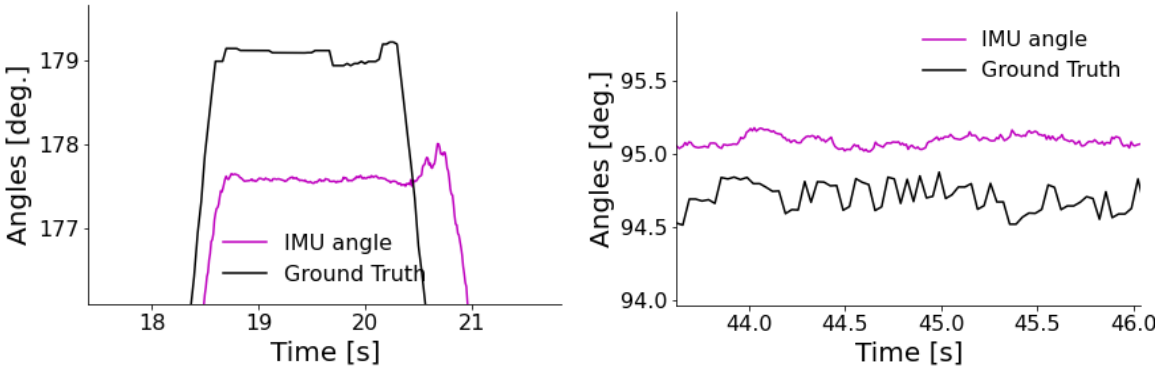


Figure 17: IMU vs Ground Truth plot.

The IMU angle model worked surprisingly well following closely the ground truth line. Taking a closer look, two observations became clear from the IMU model.

First, a small offset was observed when the modules were laying flat. This is easily seen on Figure 18(a). This offset could come from small leveling imperfections of how the accelerometer was placed on the Mori. The IMU angle model calculated a value of about 177.5° instead of 180°. An improvement to the model would add an initial calibration step to handle the offset.



((a)) Approximately 2.5° calibration offset.

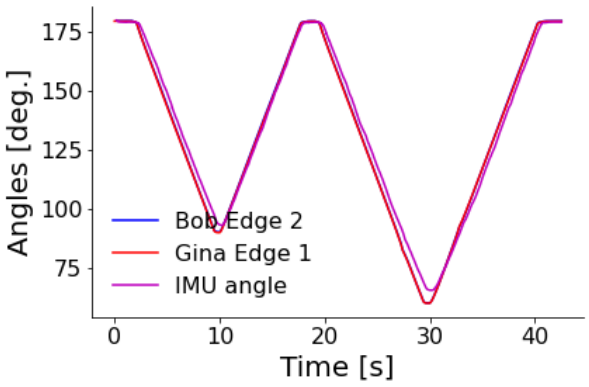
((b)) Angle difference less than 1°.

Figure 18: Small IMU deviations

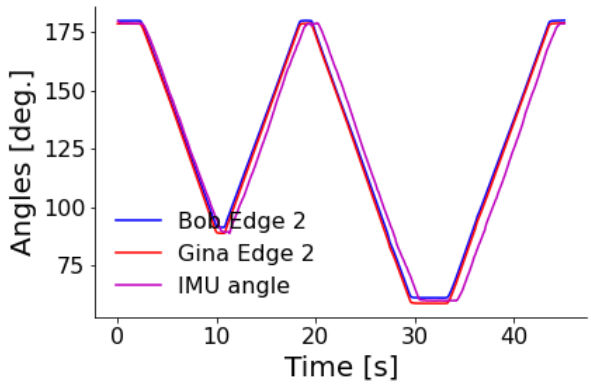
Secondly, the IMU model accurately estimated the absolute angle when one of the Mori was commanded to 90°. The motion capture software showed that the absolute angle between the

Mori was not 90°. The IMU model returned a value of 95° which very closely followed the motion capture angle of 94.5° as shown in Figure 18(b).

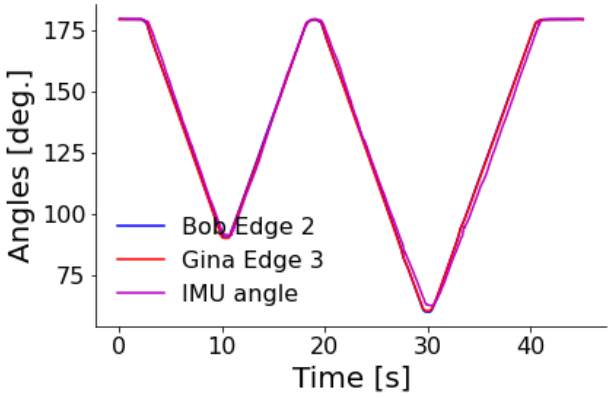
A second experiment was done in order to view how the IMU model would respond to different edge-to-edge configurations. The test consisted on cycling through the connected edge of the actively moving Mori. In this case the Mori (Gina) was tested on all of its three edges. It was tasked to perform two angle rotations to 90° and then to 60°. The results showed the model was resilient to different edge configurations as shown in Figure 19.



((a)) Edge 2 to Edge 1 test.



((b)) Edge 2 to Edge 2 test.



((c)) Edge 2 to Edge 3 test.

Figure 19: Possible edge-to-edge configurations.

4 Discussion

The accelerometer angle model offers an alternate solution to measuring the absolute angle between two modules. Furthermore, the model can be generalized to any two modules which do not have to necessarily be connected, a feature that is not possible with the angle encoder. For precise angle reading from the accelerometer, this must be programmed to a high resolution at a a high sampling rate. As previously shown the accelerometer has its own intolerances and can suffer from small calibration errors. These may result from imperfect leveling of the accelerometer upon assembly. These build imperfections could be calibrated in an initial step to account for the sensor bias.

The module was tested under different configurations. The system was cycled through different edge connections to test the model's reliability with different positioning of the accelerometer. The results showed that the model continued to work well under these conditions.

Lastly, the model was compared with Kinovea's motion-capture software. The software confirmed that the angle readings from the accelerometer were closer to the ground truth than those retrieved by the angle sensor encoder. This independent external measurement validated the accelerometer sensor model.

Future Work

The next chapter of this project would be to take everything that has been done offline and coding it into the Mori system so it can benefit from the improved sensor data in real time. This task would have to take special consideration to the different multi update rates of the Mori. Then the scalablilty of the model should be tested with multiple modules connected together. More accurate absolute angle readings between modules should result is more accurate overall inter-modular control.

A Appendix

These are honorable mentions to additional findings I made this semester.

Angle Pair Characterization

On the early weeks of the project I was a bit lost with the raw data presented to me. I thought I had to figure out which edges from which modules were together. The data was very crude and difficult to understand. The measurement files had five modules with each module recording angle orientation from three edges resulting in 15 measurements to track and little information as to which edges where connected together. This was my exploratory analysis of how I could find how many angle pairs there were in my file.

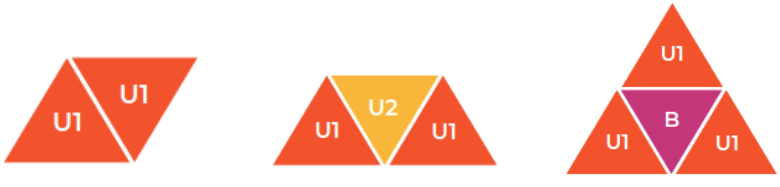
((a)) Original layout ((b)) Characterized layout

Figure 20: Own Mori characterization

When two edges meet to connect they form one angle pair. I called them an angle pair because two edges record the same angle. The initial challenge was to find out how many angle pairs there were if any configuration was possible (for example Figure 20(a)). To do this I first assumed each module carried a registry of how many edges it was connected to. This allowed me to classify each Mori into one of the three categories: **U1**, **U2**, and **B**. Moris united to one other module are classified as **U1**. Similarly, Moris united to two other modules are classified as **U2**. Lastly, Moris bounded on all sides are classified as **B** as seen in Figure 20(b). I thought that the total number of angle pairs (p) could be modeled of as a linear combination of differently bounded Mori, as described by the *angle pair equation* (eq. 9).

$$p = \alpha U_1 + \beta U_2 + \gamma B \tag{9}$$

This equation has three unknown constants to solve for $[\alpha, \beta, \gamma]$. These can easily be calculated by choosing opportune arrangements as shown in Figure 21



((a)) $N = 2$ ((b)) $N = 3$ ((c)) $N = 4$

Figure 21: Arrangements to easily solve eq. 9

For the first example we take two modules ($N = 2$) as shown in Figure 21(a). The number of

angle pairs is equal to one ($p = 1$) and the number of $U_2 = B = 0$ and $U_1 = 2$.

$$1 = \alpha(2) + \beta(0) + \gamma(0)$$

This gives us the value for $\alpha = \frac{1}{2}$.

If we then add another module ($N = 3$) as shown in Figure 21(b) we obtain the total number of angle pairs $p = 2$. This configuration has $U_1 = 1, U_2 = 2$ and $B = 0$. This configuration allow us to easily solve for β having already solved for α.

$$2 = \frac{1}{2}(2) + \beta(1) + \gamma(0)$$

This gives us the result $\beta = 1$.

Lastly, we add another module ($N = 4$) as shown in Figure 21(c). We find that the total number of angle pairs is $p = 3$ with $U_1 = 0$, $U_2 = 3$, and $B = 1$. We can than easily solve for γ, having already solved for α, and β.

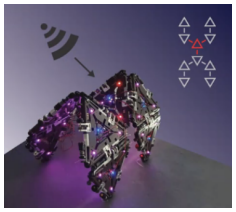
$$3 = \frac{1}{2}(3) + 1(0) + \gamma(1)$$

This gives us the result $\gamma = \frac{3}{2}$.

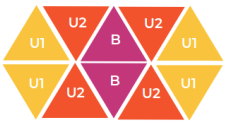
Having solved for α, β, γ, we have now formalized a generalized equation to find the number of angle pairs p in our system.

$$p = \frac{1}{2}(U_1) + (U_2) + \frac{3}{2}(B) \tag{10}$$

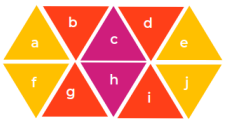
For small Mori configurations the number of angle pairs can be easily counted by sight. However, as the number of units increases this tasks becomes less trivial. This equation can be tested with any configuration; for example the Mori Dog Figure 22(a).



((a)) Mori Dog



((b)) Characterized Configuration



((c)) Letter assignment

Figure 22: Angle Pair example on Mori Dog

The Mori dog (Image source courtesy of K. Holdcroft) is made up of ten modules ($N = 10$). It has four U_1 modules, four U_2 modules, and two B modules. Utilizing the angle pair equation (eq.10) we find that this system has 9 angle pairs.

$$\frac{1}{2}(4) + (4) + \frac{3}{2}(2) = 9\, angle\, pairs$$

All possible configuration of the Mori Dog can be described by the set of these 9 angle pairs. If we assign to each Mori a letter (Figure 22(c)) we can create an Adjacency matrix called Θ formed exclusively from the 9 angle pairs listed below.

$$\vec{\theta} = \{\theta_{ab}, \theta_{bc}, \theta_{cd}, \theta_{ch}, \theta_{de}, \theta_{fg}, \theta_{gh}, \theta_{hi}, \theta_{ij}\}$$

This Adjacency matrix describes completely all shapes the Mori Dog can take.

$$\Theta = \begin{bmatrix} 0 & \theta_{ab} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \theta_{ab} & 0 & \theta_{bc} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \theta_{bc} & 0 & \theta_{cd} & 0 & 0 & 0 & \theta_{ch} & 0 & 0 \\ 0 & 0 & \theta_{cd} & 0 & \theta_{de} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \theta_{de} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \theta_{fg} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \theta_{fg} & 0 & \theta_{gh} & 0 & 0 \\ 0 & 0 & \theta_{ch} & 0 & 0 & 0 & \theta_{gh} & 0 & \theta_{hi} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \theta_{hi} & 0 & \theta_{ji} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \theta_{ji} & 0 \end{bmatrix}$$

Further mathematical analysis could add constraints to this matrix in order to determine if the desired shape is physically possible due to real mechanical limitations. Due to the nature of how the Mori are connected, all resulting adjacency matrix would be symmetrical. The relationship between the adjacency matrix and its eigenvalues and eigenvectors are further studied in spectral graph theory and could give further insights to the topology of the robotic configuration.

Magnetic Interference

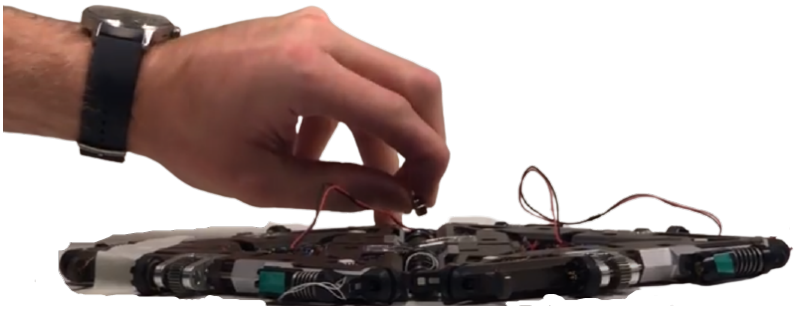
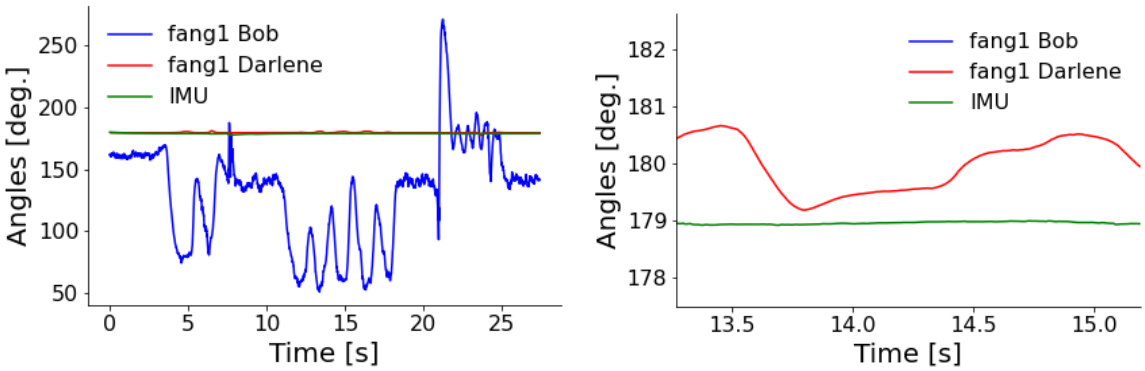


Figure 23: Angle sensor irregularity through approached magnetic field.

On a second experiment we tried to perturb the AS5048 position sensor using a neodymium magnet. The experiment consisted of connecting two Mori (Bob in blue and Darlene in Red) at an flat angle (180°). The motors were turned off in order to prevent motor damage from any angle deviations and torque compensations the wrong sensor readings could cause. The neodymium magnet was approached to the position sensor of Bob and the recording was started. Figure 24(a) shows how the strong external magnetic field disrupted Bob's perceived angle. The oscillating blue line show strong angle fluctuations despite both modules remaining completely flat. In contrast the angle readings from both Darlene (red) and IMU (green) remained largely constant around 180° as shown in Figure 24(b).



((a)) Visible magnetic perturbation.

((b)) Accelerometer and Darlene angle values.

Figure 24: Induced angle deviation via magnet.

This experiment shows how adding redundancy to the system creates fail-safes in the case one or two angle sensors become defective.

References

[1] J. Seo, J. Paik, and M. Yim, "Modular Reconfigurable Robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, pp. 63–88, 5 2019.

[2] K. Støy, "Reconfigurable robots," *Technology Review*, vol. 105, no. 4, p. 54, 2002.

[3] C. Parrott, T. J. Dodd, and R. Groß, "HiGen: A high-speed genderless mechanical connection mechanism with single-sided disconnect for self-reconfigurable modular robots," *IEEE International Conference on Intelligent Robots and Systems*, no. Iros, pp. 3926–3932, 2014.

[4] H. Kurokawa, A. Kamimura, E. Yoshida, K. Tomita, S. Murata, and S. Kokaji, "Self-Reconfigurable Modular Robot (M-TRAN) and its Motion Design," *Proceedings of the 7th International Conference on Control, Automation, Robotics and Vision, ICARCV 2002*, pp. 51–56, 2002.

[5] M. W. Jørgensen, E. H. Østergaard, and H. H. Lund, "Modular ATRON: Modules for a self-reconfigurable robot," *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 2, pp. 2068–2073, 2004.

[6] A. Spröwitz, R. Moeckel, M. Vespignani, S. Bonardi, and A. J. Ijspeert, "Roombots: A hardware perspective on 3D self-reconfiguration and locomotion with a homogeneous modular robot," *Robotics and Autonomous Systems*, vol. 62, pp. 1016–1033, 2014.

[7] J. Davey, N. Kwok, and M. Yim, "Emulating self-reconfigurable robots - Design of the SMORES system," *IEEE International Conference on Intelligent Robots and Systems*, pp. 4464–4469, 2012.

[8] C. Heinrich, "From Modular Origami Robots to Polygon-based Modular Systems : a New Paradigm in Reconfigurable Robotics," 2020.

[9] "Robot pebbles: One centimeter modules for programmable matter through self-disassembly," 2010.

[10] N. Semiconductors, "MMA8452Q 3-axis, 12-bit/8-bit digital accelerometer," 2016.