Bar-Ilan University

Faculty of Life Sciences

Final Project in Computational Biology

A non-Newtonian Fluid Robot

Submitted by Nomi Hadar

ID: 308043629

Supervisor: Dr. Ido Bachelet

September 2015

Table of Contents

Introduction

Robots are extensively employed in diverse fields of technology such as industry, medicine, security and research. Robots are able to do jobs which are difficult, dangerous or impossible for humans, and today they are an essential part of life. Recently, a particular interest has been shown in unconventional robots inspired by organisms such as worms, squids and hovering insects, which exhibit significant fitness to certain settings and tasks.
In this project I was integrated into a research team that explored a novel concept in unconventional robotics - a robot made of non-Newtonian fluid (NNF). These fluids, defined by a nonlinear relation between viscosity and strain rate, provide an interesting framework for producing animal-like behaviors as well as non-animal features such as splitting and plasticity. My job was to quantify the robot behavior by video analyzing of experiments in which it performs several tasks.

1. Background

2.1 Non-Newtonian fluid

A non-Newtonian fluid is a fluid whose viscosity is variable based on applied stress or force. Many polymer solutions and molten polymers are NNFs, as are many commonly found substances such as ketchup, starch suspensions, paint, and shampoo. If you poke NNF such as cornstarch with your finger and apply a large force, it becomes very viscous and behaves like a solid. If you gently pour it, applying little force, it will flow like water.
To define it scientifically, a NNF is a fluid whose flow properties are not described by a single constant value of viscosity. In a Newtonian fluid like water the relation between the shear stress and the strain rate is linear, the constant of proportionality being the coefficient of viscosity. Therefore the behavior of Newtonian fluids can be described exclusively by temperature and pressure. In a NNF, however, the relation between the shear stress and the strain rate is nonlinear, and can even be time-dependent. Thus, a constant coefficient of viscosity cannot be defined. This means that the physical behavior of NNF depends on the forces acting on it from second to second.

2.2 Different types of non-Newtonian fluids

Not all NNFs behave in the same way when stress is applied – some become more solid, others more fluid. Some NNFs react as a result of the amount of stress applied, while others react as a result of the length of time that stress is applied.

The table below summarizes four types of NNFs:

Type of behavior	Viscosity	Example
Thixotropic	Decreases with stress over time	Honey – keep stirring, and solid honey becomes liquid
Rheopectic	Increases with stress over time	Cream – the longer you whip it the thicker it gets
Shear thinning	Decreases with increased stress	Tomato sauce
Dilatant or shear thickening	Increases with increased stress	cornstarch dissolved in water

In this work we used **dilatants**, as they are capable, as mentioned, to change their behavior in accordance to the amount of stress being put on them.

2.3 Controlling non-Newtonian fluids by acoustic force

When placing dilatant NNF on a vibrating speaker, it demonstrates other unusual properties, like protruding "fingers" and holes that persist after creating them (figure 1). The speaker applies stress in varying intensities, rapidly changing the viscosity of the fluid and putting it in an odd middle ground between a liquid and a solid.

1: NNF on active speaker

When I have joined the project, my research team have already designed and built a system that takes advantage of the acoustic force to drive the robot. This system is consists of a 2D array of acoustic transducing speakers (figure 2). A canvas is stretched over the array to create a planar surface and the robot is placed on top of it (figure 3). The parameters of each speaker (volume, frequency) can be controlled separately to allow maximal array flexibility. A monitoring system is based on video cameras positioned in 3 axes.

2: speakers array

Their results clearly showed that it is possible to control NNF robots by acoustic force and make them perform tasks such as moving around, carrying cargo, shape shifting, splitting and more.

3: NNF robot on the speakers-array surface.

2.4 Benefits of robots made of non-Newtonian fluid

Dilatant NNF provide an interesting framework for producing animal-like behaviors in matter, but moreover, can exhibit features that are impossible for organisms to exhibit. For example, a fluid robot could split into two or more separate robots, each carrying out a different task, than aggregate back to form the whole robot. Splitting behavior also includes percolation through porous media as a discontinuous body, enabling dripping through sand or debris.

In addition, as all living things robots are still bound to the same physical boundaries such as a membrane, a "skin" or other casing or chassis. A fluid robot, conversely, could exhibit enormous plasticity constrained only by volume of matter, enabling drastic changes in its length, surface area, or the space it occupies.

By combining animal and non-animal behaviors, a fluid robot would be able to carry out a wide range of tasks that are impossible to any other machine or organism.

2. Aim and Methods

The general aim of this study was to describe the design, properties and behavior of an NNF-based robot. Specifically, my job was to evaluate and quantify the robot behavior by processing the videos that documents the experiments (tasks that the robot performs) using algorithms for image analysis and 2/3D information extraction. The average length of such video is three minutes.

Methods: MATALB image and video processing toolbox.

3. Significance

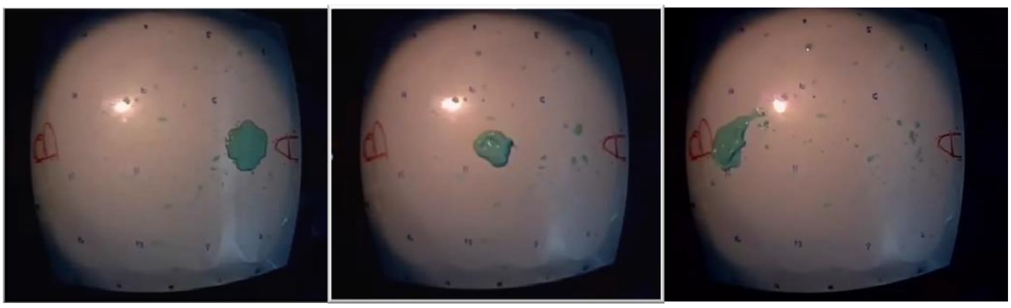
This study is a doorway to a novel field of robotics, enabling us to make robots with functions never seen before. In addition, this study makes it possible to research the behavior of mobile formless bodies, as well as significantly increasing the knowledge we have on NNF, which the physics behind it is not fully understood. Finally, this study holds possibilities to design and engineer fully working liquid robots in the near future, and its applications are both numerous and beneficial.

4. Outputs and Results

In this section I will first demonstrate on a single video how we can mark the robot in order to be able to track it in each second. Then I will discuss what data, such as velocity and area, can be extracted after the robot is marked. Finally I will show more examples and what can be taught from them on the robot's behavior.

4.1 Marking the NNF robot

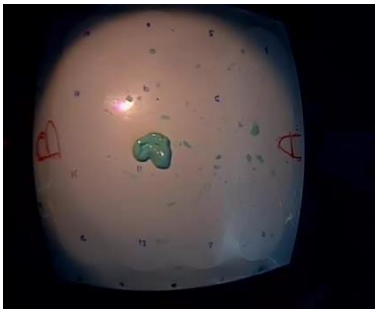
The first challenge was to take a video in which the robot moves from point A to point B (figure 4) and mark the robot's outline and center.



4: Snapshots from a video of A -> B movement

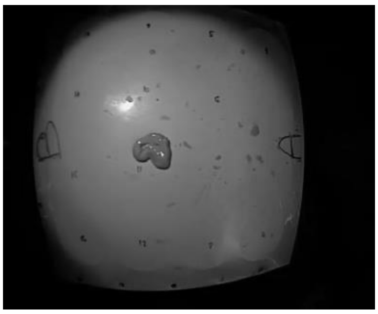
We will focus on a single frame to demonstrate, step by step, the actions required to isolate the robot from its background. Video is a collection of frames, so once we mark the robot in a single frame, we can apply the same actions to all other frames.

Original frame:



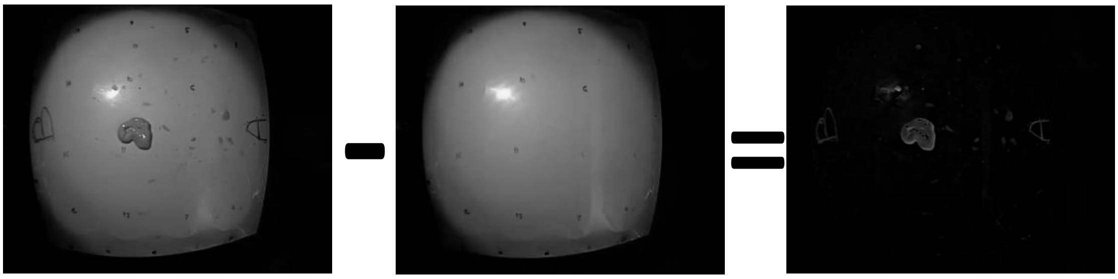
Step 1 – Converting to WB:

Because an RGB image weights three times more than WB image (a RGB image is represented by three matrixes vs only one matrix for a WB image), while the color has no meaning, we will convert the frame to WB.



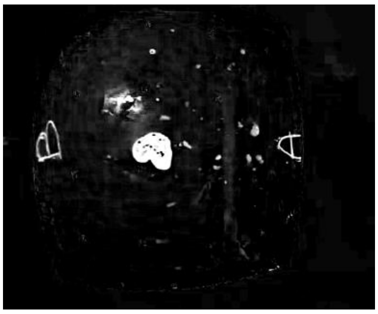
Step 2 - Subtracting background:

WB images are actually matrixes with values ranging from 0 (which represents the black color) to 255 (white). Therefore we can perform on them mathematical operations such as addition and subtraction. If we subtract a clean image of the canvas (without the robot on the top of it) from an image with the robot, we get the following result:



Pixels with same or close values became 0 or close to 0, i.e. they became black. The areas that didn't become black are only those which differ between the images, such as the area of the robot.

Step 3 – Increasing contrast:



The resulting image from the second step is now a bit dark for analysis, so now we used a Matlab function to increase the contrast. This highlighted the robot but also reduced the uniformity of the background.

Step 4 – Converting to binary image:

A binary image is a digital image that has only two possible values for each pixel. Matlab offers function which convert image to binary image, based on threshold. The output image replaces all pixels in the input image with value greater than level with the value 1 (white) and replaces all other pixels with the value 0 (black). This conversion is convenient when we want to distinguish between an object and its background. In addition, binary images are the input for many of the Matlab toolbox functions for image analysis.

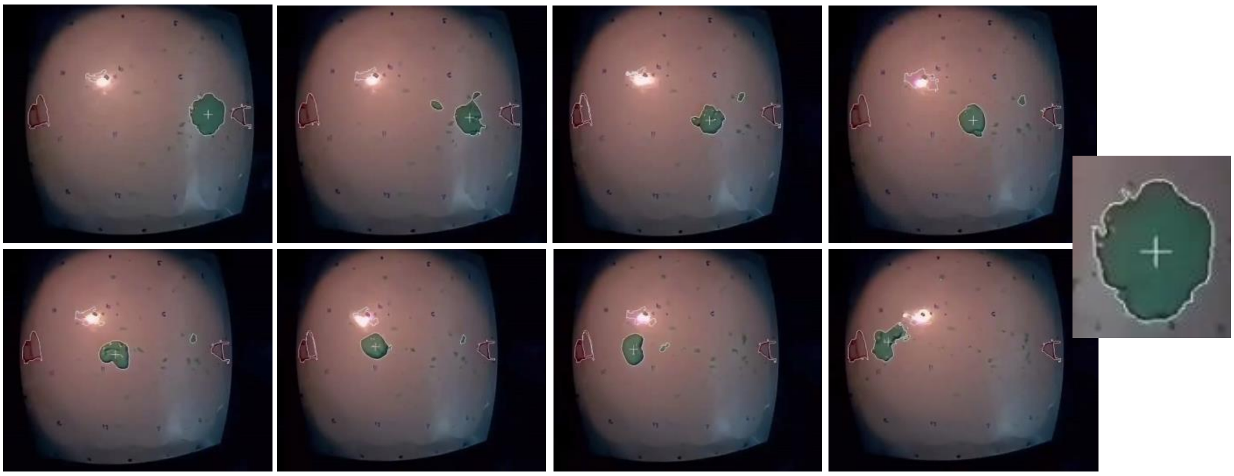
Step 5 – Cleaning noise and filling holes:

We applied on the image two more Matlab functions. Both get a binary image and a maximal number of pixels, and operate on areas that have fewer than the given number of pixels. The first removes small objects and thus cleans the image, while the second fills small holes in the big objects.

Step 6 and final - Drawing outlines:

In the final stage we drew the exterior boundaries of the objects, and mark the centroid of the robot (not shown here).

When applying those actions to all frames we get the following result:



5: Snapshots from the output video of A -> B movement

4.2 Calculating center, area and velocity

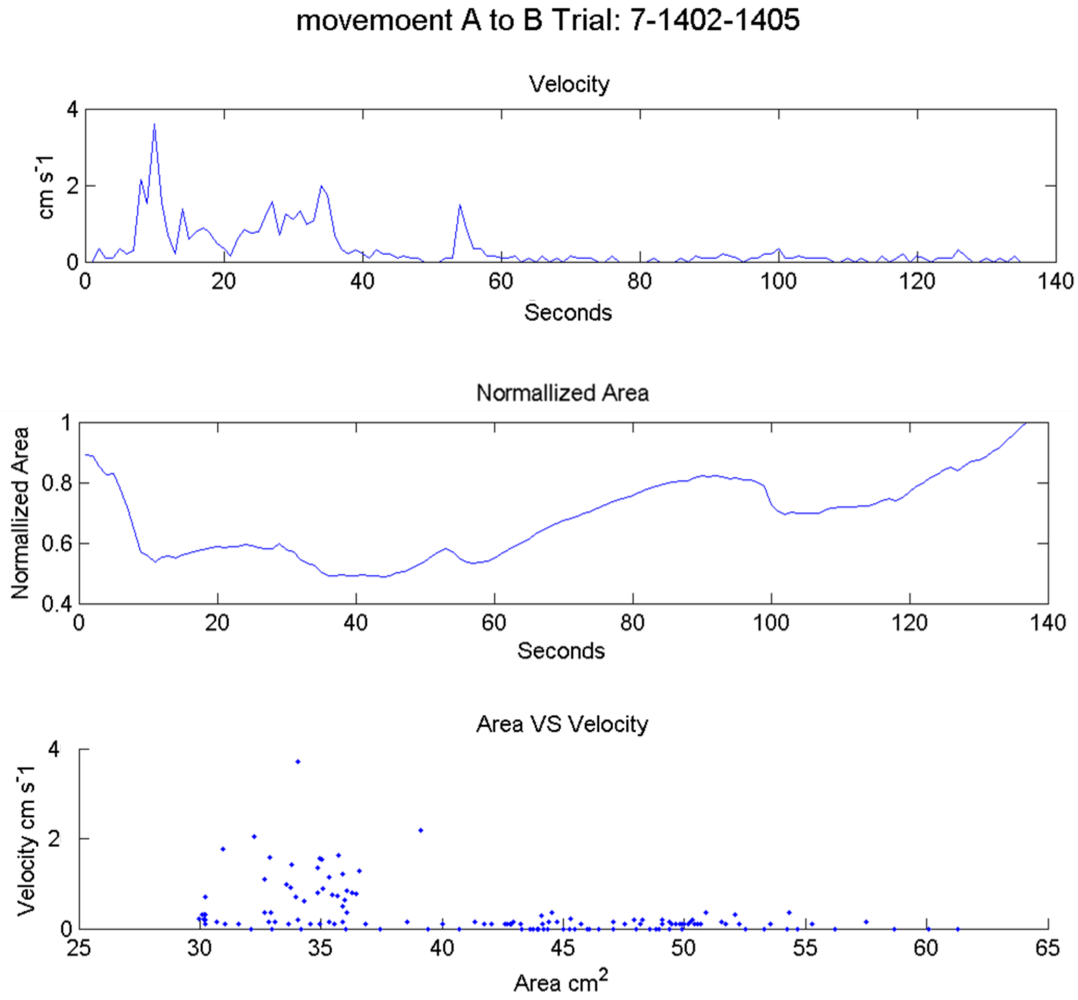
After detecting the robot we were able to use MATALB image processing toolbox built-in functions to find its *center* and its *area* in each frame. Those functions get a binary image of objects and return the objects' centers in a 2D coordinate, and their area (in pixels). In order to convert to cm we measured the length of the surface and calculate the ratio between it and its number of pixels in the video.

For calculating *velocity per second* we sampled two centers from two consecutive seconds and calculated the Euclidean distance between them. For calculating *area per second* and *center per second* we averaged the areas the centers, respectively, obtained for an entire second. This appears in the following pseudo-code:

```
Centers contains spot's center (2D coordinate) in each frame
Areas contains spot's area (in pixels) in each frame

Set FPS (frames per second) to the frame rate of the video
Set counter to 1

While counter is less than video's number of seconds

    % set frames indexes to the start and the end of the current second
    %i.e if FPS = 25 and counter = 2 then start index = 26 and end index = 50
    start-index             <- (counter-1)*FPS +1;
    end-index               <- (counter)*FPS;

    % calculate velocity (pixels/sec)
    current-coordinate      <- Centers[start-index]
    next-coordinate         <- Centers[end-index + 1]
    distance                <- Euclidean distance between current and next coordinates
    save distance for current second

    % calculate average centers
    average-center          <- average of Centers[from start-index to end-index]
    save average-center for current second

    % calculate average area
    average-area            <- average of Areas[from start-index to end-index]
    save average-area for current second

    counter                 <- counter + 1;

    End loop
```

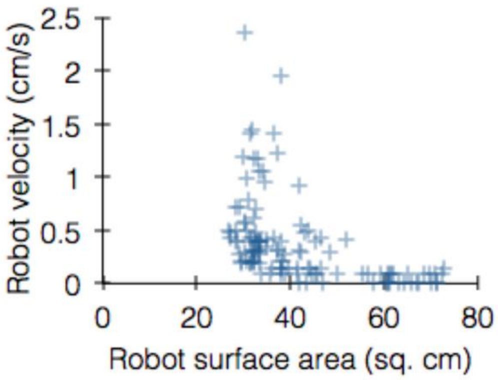
Once we have this data we can display it on graphs, to answer questions about the robot's behavior such as whether there is a correlation between the surface speed of the robot, what is its average and maximal speed, etc.

Next page provides several examples.

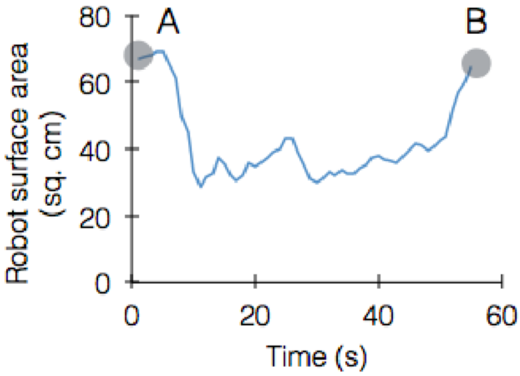


6: graphs for one of the A -> B movement trials. Graph 1 – velocity. Graph 2 – normalized area (area divided by maximal area). Graph 3 – area vs velocity.

The bottom left graph displays area vs velocity from another trial. It can be seen clearly that the higher the speed of the robot, the smaller its area. In other words, the robot is compressed to increase speed.



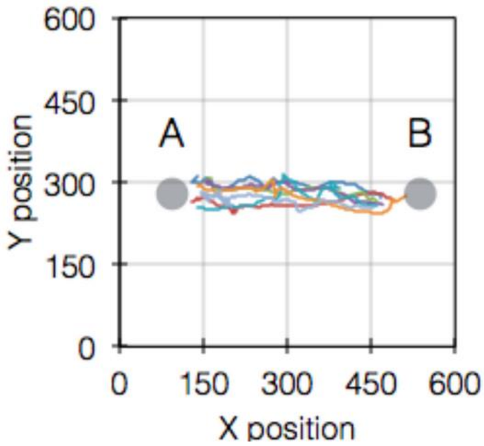
7: Correlation between robot surface area and velocity. Each point represents a second in a typical experiment



8: Robot surface area (in sq. cm) change along a path from A to B in a typical experiment.

4.3 Reproducibility

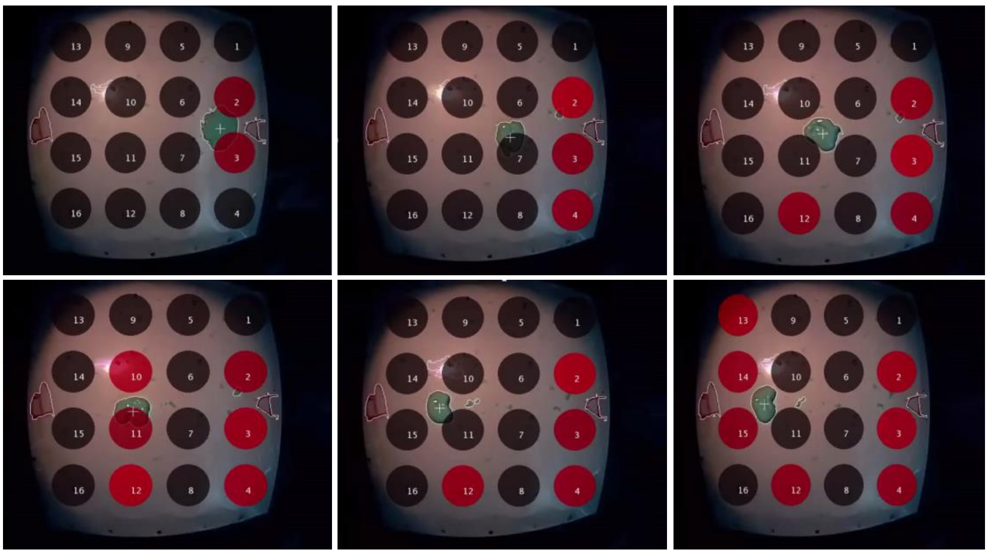
Reproducibility is the ability of an experiment to be duplicated. The graph on the right displays the paths of the robot in its way from point A to B in 10 independent experiments, and shows that this experiment can be replicated.



9: Paths of a fluid robot moving from point A to B in 10 independent experiments. X and Y positions are in mm.

4.4 A visual representation of speakers' activity

To make it easier to follow which speaker is active at any moment, we integrated in the videos a visual representation of the speakers. Each speaker is represented by a circle in proximity to its original location as it appears in the video. An active speaker appears in a red color in a shade relative to the speaker intensity. The red shades are divided into 10 levels. Assumed maximal speaker intensity of 900V, thus every red level represents 90V. The higher the intensity, the stronger the red color.



10: Snapshots from the output video of A -> B movement with speakers visualization

Knowing which speaker has been active allows us to test the influence of the intensity / frequency of the speakers on the robot's behavior. In addition it allows us to replicate an experiment precisely and furthermore to build a program that gets a list of speakers per second and activates the speakers automatically without human intervention.

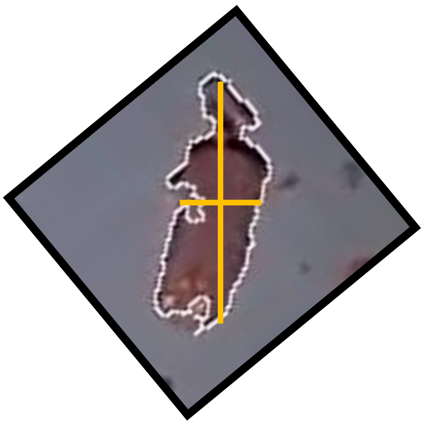
4.5 Aspect ratio

The aspect ratio of a geometric shape is the ratio of its sizes in different dimensions. The aspect ratio of a circle, for example, is 1. In the following experiment the robot is transformed from a circle to an ellipse.

11: Snapshots from a transformation experiment

The task was to calculate the aspect ratio of the robot during its transformation, i.e. the maximal ratio between its height and its length. The aspect ratio was calculated by the following algorithm:

```
initialize max-ratio to 0
set degrees to 1

    while degrees is less than or equal to 90, increased by 3

    rotate picture at current number of degrees
    create a cross in the center of the robot  %center was calculated earlier
    % the cross is always straight, while the image is rotating

    horizontal-line   <- rotated image ∩ horizontal-cross-line
    vertical-line     <- rotated image ∩ vertical-cross-line

    if        vertical-line is longer  than horizontal-line
              temp    <- vertical-line divided by horizontal-line
    else
              temp    <- horizontal-line divided by vertical-line
    end if

    if        temp is bigger than max-ratio
              max-ratio         <- temp
    end if

    degrees           <- degrees + 3;

    end loop
```



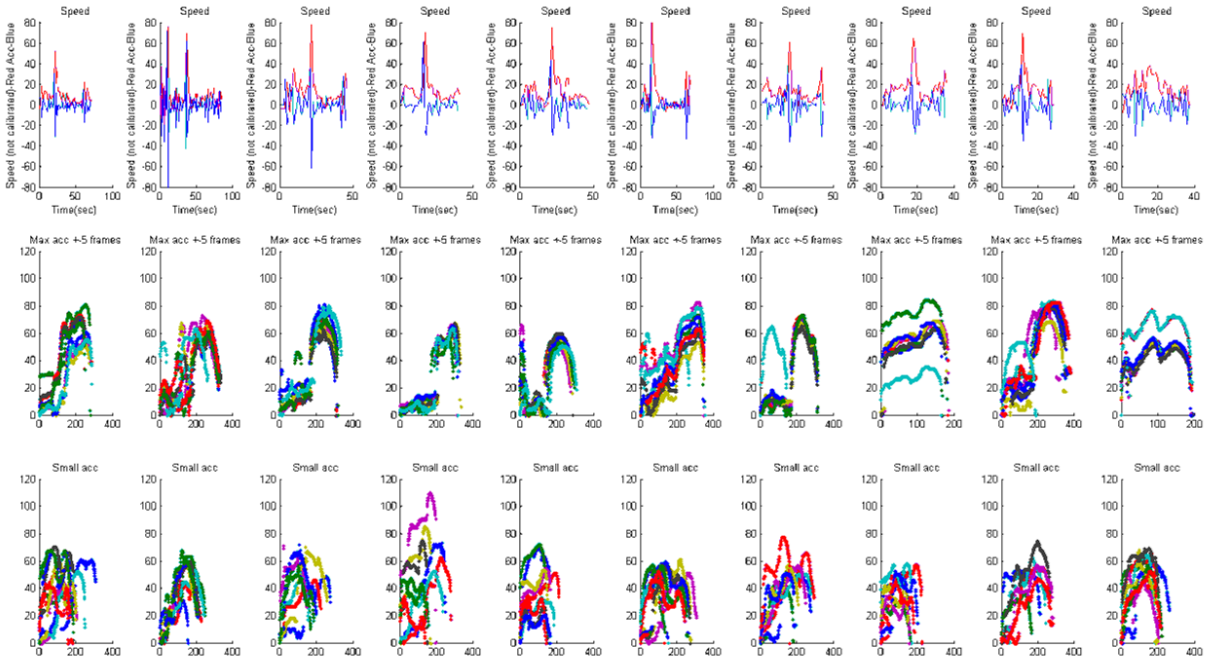
12: Demonstration of the algorithm

4.6 Robot hump outlines at high vs. low acceleration

The following experiment examines the robot hump outlines (camera angle is from the side rather than from top).

To examine what factors affect the hump we checked the speed and acceleration as shown below.

13: Hump



The following graphs are a summary of 10 independent movement (A to B) experiments. Each column is an experiment, while the rows are ordered as follows:

Top, velocity (red) and acceleration (blue) per frame.

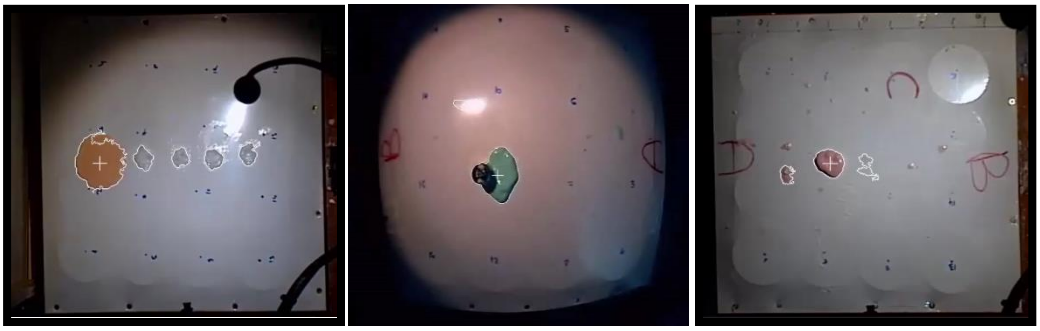
Middle, robot outline (pixels) at maximum acceleration frame +/- 5 frames (overlaid together).

Bottom, robot outline (pixels) at an average acceleration frame +/- 5 frames (overlaid together).

The results show that the size of the acceleration affects the shape of the hump. Robot hump behavior is clearly shown in middle row samples, at high acceleration, while behavior is more random in the bottom row samples, at low acceleration.

4.7 More outputs

We ran the algorithms introduced in the previous sections on many repeated trials, as well as on additional experiments such as moving from A to B to C, a cargo experiment in which the robot carries weight and a counting experiment in which the robot counts to 3.



14: Left: counting experiment, middle: cargo experiment, right: A -> B -> C movement experiment.

5. Summery

Video analysis is a powerful tool which allows us to discover things that cannot be seen just by looking at the photographed object. In this study video analysis allowed us to track the movement of the robot, as well as calculating its area, velocity, acceleration, aspect ratio and more. Without it, it was difficult to say something about the behavior of the robot.

This study which is a part of the NNF Robot research was accepted to the Artificial Life journal.

A non-Newtonian fluid robot

Guy Hachmon[1], Noam Mamet[1], Sapir Sasson, Larisa Zlatin, Sivan Friedman, Tal Barkai, Nomi Hadar, Almogit Abu-Horowitz, Ido Bachelet[2]

Faculty of Life Sciences and Institute of Nanotechnology & Advanced Materials, Bar-Ilan University, Ramat Gan 52900, Israel.

1. These authors contributed equally to this work
2. Author for correspondence. Address: Faculty of Life Sciences, Bar-Ilan University, Ramat Gan 52900, Israel. Phone: +972 3 7384312. Email: ido.bachelet@biu.ac.il

New types of robots inspired by biological principles of assembly, locomotion and behavior have been recently described. In this work we explored the concept of robots that are based on more fundamental physical phenomena, such as fluid dynamics, and their potential capabilities. We report a robot made entirely of non-Newtonian fluid, driven by anisotropic shear strains created by spatial patterns of audio waves. We demonstrate various robotic primitives such as locomotion and transport of metallic loads – up to 6-fold heavier than the robot itself – between points, splitting and merging, shapeshifting, percolation through gratings, and counting to 3. We also utilized interactions between multiple robots carrying chemical loads to drive a bulk chemical synthetic reaction. Free of constraints such as defined outline or skin, fluid robots could adapt more easily to unfamiliar, hostile, or chaotic environments and carry out tasks that neither living organisms nor conventional machines are capable of.

Recent developments in robotics and orthogonal fields such as materials sciences and engineering have led to a paradigm shift towards unconventional robots inspired by living organisms, their properties and capabilities (1-6). Such robots are highly adaptable for tasks in challenging environments. For example, soft, solid-state robots fabricated from elastic polymers and devoid of hard parts or joints could potentially withstand high mechanical loads and survive crushing and bending forces lethal to conventional machines.

However, the living organisms that inspire these robots are still largely constrained by structural and functional biological barriers. For example, skins and exoskeletons, which protect the integrity and chemical composition of organisms, substantially limit their morphological flexibility. Moreover, even organisms devoid of skeleton (*Mollusca*) tend to adhere to well-defined outlines and shapes, and while resilient to bending and crushing, they are highly vulnerable to cutting and shearing forces. These issues could impair the survivability of robots designed to operate in highly dynamic or hostile environments in which the conditions cannot be predicted a-priori. But they could be addressed by shifting to nonliving matter as the source of inspiration for robots with a distinctly non-biological set of behaviors and capabilities. In this work we focused on fluid dynamics as a paradigm for this challenge.

15: The first page of the article

6. References

- https://en.wikipedia.org/wiki/Non-Newtonian_fluid
- http://www.ge.com/press/scienceworkshop/docs/pdf/Non_Newtonian_Fluid_with_Standards.pdf
- Search proposal of Guy Hachmon, a M.Sc student and the team leader
- Matlab tutorial:
  http://www.mathworks.com/academia/student_center/tutorials/launchpad.html
- Basic Image Enhancement and Analysis Techniques / Matlab tutorial:
  http://www.mathworks.com/help/images/image-enhancement-and-analysis.html

7. Attachments

- ***A non-Newtonian fluid robot*** - the article
- Scripts used in this study – Matlab files (.m extension)
- Several original and edited video files of the experiments

Signature of supervisor: