



Modular and Flexible Tangible Molecular Interface for Interactive Molecular Simulation Based on Internet of Things Approach

Bastien Vincke¹(✉), Mohamed Anis Ghaoui¹, Nicolas Férey²,
Xavier Martinez^{2,3}, and Loïc Brochot¹

¹ SATIE - CNRS UMR 8029, Univ. Paris-Sud, Université Paris-Saclay,
Digiteo Labs, BAT 660, 91405 Orsay Cedex, France

bastien.vincke@u-psud.fr

² VENISE Team, LIMSI, CNRS, Univ. Paris-Sud, Université Paris-Saclay,
Orsay, France

³ Laboratoire de Biochimie Théorique, CNRS, UPR9080,
Univ Paris Diderot, Sorbonne Paris Cité, PSL Research University, Paris, France

Abstract. Rational Drug Design, is an approach based on detailed knowledge of molecular interactions and dynamic of bio-molecules, focused on rationally constructed new drugs, as opposed to conventional and empirical approaches. These Rational Drug Design approaches involve designing new digital and interactive tools that take into account unconventional physical laws at nanoscopic scale using molecular simulations. However, many advanced interactive devices, such as haptic devices for manipulating and steering molecular models *in silico* especially to support protein docking activities, are not well-adapted to consider the numerous degrees of freedom and flexibility of bio-molecules. In this paper, we propose to address this issue by implementing an innovative approach benefiting from a physical, modular and articulated molecular model augmented by an Internet Of Thing approach, to create, design and steer its *in silico* twin model through a tangible molecular interface.

Keywords: Virtual reality · Tangible interface · Human-Machine Interface · Molecular simulation · Internet of Things · Rational Drug Design

1 Introduction

Rational Drug Design requires a detailed understanding of how proteins work. The role of each protein depends on many factors: its three-dimensional structure, its physico-chemical and dynamic properties and bio-mechanical behaviour, and its interactions with other partners. These features, particularly dynamic,

Supported by Farman Institute (through the project BIOTIN'IT).

© Springer Nature Switzerland AG 2019

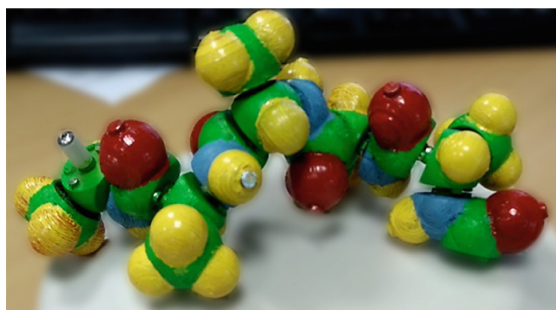
P. Bourdot et al. (Eds.): EuroVR 2019, LNCS 11883, pp. 24–32, 2019.

https://doi.org/10.1007/978-3-030-31908-3_2

are difficult to access through experimentation, which is why numerical simulation tools have been designed to obtain theoretical results on protein dynamics. Nowadays, molecular simulations are very efficient and allow to visualise and interact with the simulated molecules, opening new perspectives to build, simulate and test in interactive time the impact of proteins and more generally molecules entirely designed by experts.

However, one of the major obstacles to the Rational Drug Design approach lies in the limitations of Human-Machine Interfaces for manipulating complex, three-dimensional, flexible objects during simulations.

The use of interactive interfaces began in the 1970s. The haptic interfaces used were excessively expensive and voluminous. Although they are now available at low cost and can be used daily in an office or virtual reality environments, these interactive molecular simulation techniques show their limitations, in particular for dynamic manipulations and therefore flexible virtual protein models, mainly due to the limits of the 6 degrees of freedom at the input of most 3D interaction devices. To overcome this problem, Arthur Olson's team has conducted research [1,2] to transform physical models into more tangible and richer interfaces than traditional interaction devices (keyboard, mouse), to manipulate virtual molecular representations combined with augmented reality techniques. The approach was using an explicit marking system (ARToolkit) or retro-reflective infrared markers.



(a) Peppyptide interface proposed by [3]



(b) New assembled model

Fig. 1. Model samples comparison

The use of markers is often problematic. And thus, Xavier Martinez proposed a reliable, robust, non-intrusive and marker-free approach to reconstruct a three-dimensional digital molecular model from the modular and flexible physical object manipulated in interactive time [4]. The approach is based on an external camera that poses a major occlusion problem (Fig. 2). Jacobson proposed a generic tangible interface to control

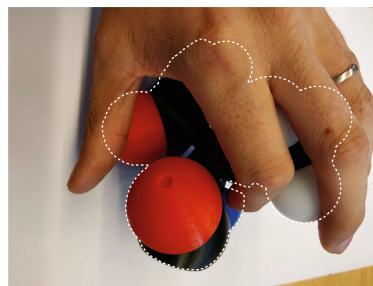


Fig. 2. Occlusion example

articulated virtual characters [5]. However, its interface uses wired connectivity, no automatic pairing and is not suitable for protein manipulation.

2 Contributions

We have designed an instrumented tangible molecular interface (Fig. 1b, allowing the manipulation of a modular and flexible molecular physical model to manipulate a digital molecular avatar. Our interface is based on a modular and flexible 3d model called Peppytide (Fig. 1a). This physical model of the protein, which has shown its biological relevance [3], is a rigid body chain including several atoms, assembled by mechanical bonds whose degrees of freedom are summarised in a sequence of torques of rotation angles for each amino acid (ϕ , ψ). Indeed, a protein conformation can therefore be reconstructed in 3D only from this sequence of angles. We have therefore instrumented the Peppytide model to allow real-time 3D protein reconstruction.

3 Proposed Interface

Several modifications were made to the model to allow for reconstruction. We have modified the structure of the model to accommodate an electronic board equipped with an angle sensor, a battery and a wireless communication system.

3.1 Mechanics

First, we had to double the dimensions of the Peppytide model. We transitioned from a scale of 93000000 ($1\text{\AA} = 9.35\text{ mm}$) to a scale of 186000000 ($1\text{\AA} = 18.7\text{ mm}$) compared to reality. To avoid some mechanical constraints, the Van Der Waals atom radii were diminished by multiplying it by 0.7, then scaling them up to 186000000. The pivot junctions have been modified and adapted to the implied constraints by embedding electronics. We created a shaft plugged in a hollow tube to allow its rotation. The tip of the shaft has an indentation that acts as a socket for a magnet (used for the angular sensor) to be placed in. With this method, the pivot junctions will not disturb the embedded electronics. On the plinths of these junctions, we kept the magnets of the Peppytide model to simulate potential peaks.

The integration of the electronics can not be done without accessing the inside of the pieces. To do that, we developed a system of assembly of atoms after 3D printing them (Fig. 3). On each atom, there are 2 shafts and 2 holes. The assembly is performed by simply encasing the two pieces. This method allows both the placement of electronic components inside the atom and printing the atoms in conventional colours making them easily distinguishable by the user. Finally, on the electronic board, there are many components that have to be visible from the outside. We must add sockets in the pivot junctions.

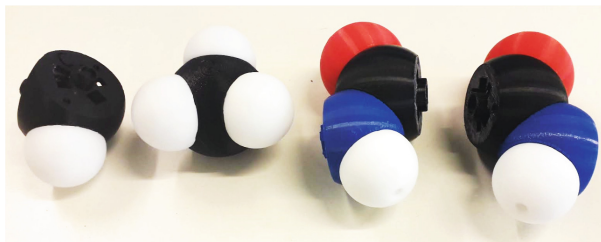
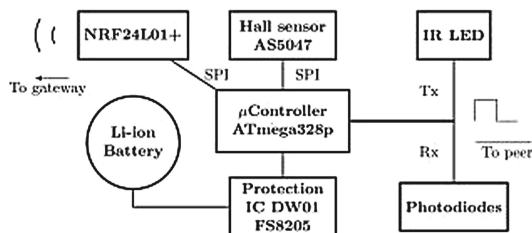


Fig. 3. Colour-coded molecular models

3.2 Hardware

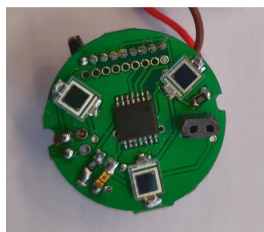
Each of the previously mentioned pivot junction is equipped with the following set of electronic devices to “bring it to life”, allowing it to be interactive and connected to a gateway μ controller (Fig. 4). The following block diagram summarises all the embedded electronics:

- *ATmega328p* μ C [6].
- *NRF24L01+* [7].
- IR LED.
- 3 Photodiodes.
- *AS5047P* Magnet rotary position sensor.
- Protection IC : DW01 FS8205.
- Lithium-Ion 3.6V Battery 150mAh.

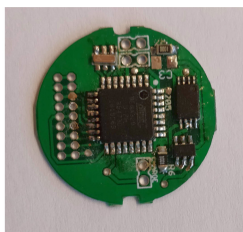


Block diagram : Embedded electronics

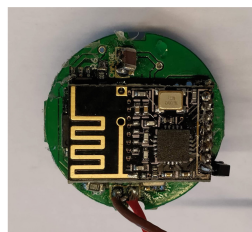
Micro-controller. The main chip is an *ATmega328p* μ controller that interfaces its surrounding modules either by SPI or digital pins. This μ controller is cheap, low power, robust and widely supported by the Arduino community with all the required libraries. Thus, it is the most suitable choice for these prototypes.



(a) Front



(b) Back



(c) Back with RF module

Fig. 4. Electronic board implemented at each junction

Angular Measurements at Joints. Angle measurement is performed using an *AS5047P* hall effect sensor. This kind of sensor allows a very accurate (14bits) and non-contact angle measurement. The axis is equipped with a diametrically polarized magnet. The sensor is placed in front of this axis and is interfaced using an SPI bus.

Wireless Interface. Each molecule has to communicate with the base gateway through a RF module, the *NRF24L01+* [7]. This module is a low power, cheap and efficient wireless chip that allows the construction of a wireless network. Even though this module can only perform duplex communication, its communication rate is more than enough to make it seem full-duplex and to allow interactivity between the simulator and the end user. It is interfaced with SPI bus and has many open-source libraries to support it. To ensure minimum energy consumption, the RF module are set to the lowest amplification level and data rate available (-18 dBm, 250 Kpbs).

Automated Molecular Pairing. In our prototype, Every molecule part junction has its dedicated *NRF24L01+* chip and an unique associated identifier (ID). To be able to detect the current molecular topology, each peered junction has to know with which other node it is peered with. That step is performed by Timer interrupt asynchronous communication through the use of an IR LED and photo-diodes to exchange the unique IDs (Fig. 5).

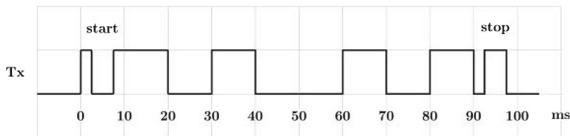


Fig. 5. Tx line visualised: sending 0xA5 (LSB to MSB)

3.3 Embedded Software

Bootloader. The bootloader is a start-up program that is called with each reset of the *ATMega328p* chip before the main program. The default bootloader for this chip provided by Arduino [8] is setup to listen to the serial port for 4 seconds and intercept the incoming bytes, if there is any, it locks itself into update mode. The intercepted byte-stream consists of a compiled code for the new main program to be overwritten in the flash memory and then executed.

Low-Level Library. Choosing the suitable components for this project was a very selective process because each component had to satisfy many criteria: low

power, reliability, long life expectancy and suitable support. Every component that wasn't supported by a library had to be interfaced by a customised program of ours. The RF module uses the NRF24L01+ library [9] and SPI [10]. The topology presented next is setup with the MySensors Library [11].

Wireless Protocol. The chosen wireless module has quite a really constraining downside. It can only effectively connect up to 6 others of its kin [12]. To get pass this limitation, a tree topology must be implemented and allow some modules to act as routers for their children. Each node has a known address and uses it to receive information from the network [13]. This allow for a total of 3125 nodes in a static network. The main problem with this static topology is: if one of the nodes just disconnects (for any reason), all of its children lose connection to the Master-node. Thus, for our application, it is mandatory to setup a dynamic topology and enable emergency rerouting for each node. And so, each node communicates using a default address to request a new address from the Master-node [11]. If one router node has to disconnect, its children can re-route to other available nodes or promote themselves in the topology. This renders the network immune to sudden death (Fig. 6).

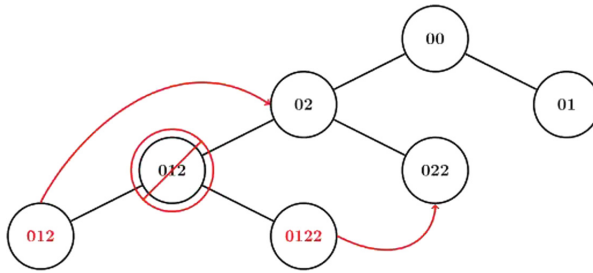


Fig. 6. Dynamic tree graph topology

Over the Air Programming. During the preparation of some prototypes, one difficulty was that the reprogramming of the μ controller inside the module was not possible without tearing the part to pieces. To overcome this, we implemented a modified bootloader [14] that was OTA-enabled [15]. This means that after rebooting the chip, the bootloader would listen for 4 seconds to incoming bytes either from the serial port or the RF module. If any data was available, the bootloader locks itself in update mode. It can retrieve the updated program by requesting it from gateway. The data coherence is ensured by the bootloader requesting the new firmware page by page, 16 bytes per page and will act in burst mode trying to get as much pages as possible in succession. The watchdogs forces a page request every 4s ensuring a minimal update speed of 16 bytes/4s and preventing any possible bricking.

Base Gateway. The gateway is the Master-node of network and is always connected to the computer first. It is always identified by the ID 0 and is responsible for the address attribution and firmware update for each node requesting it. The gateway also proceeds to output some serial data stream that is available to be read through the serial port with a python script that reinterprets the basic string into a more evolved OSC commands for the simulator that is acting as an OSC server.

3.4 Simulator

The implementation of the rendering and the physic simulation is done in Unity3D that provides both a high-performance rendering pipeline combined with a reliable physic engine (Nvidia PhysX).

Two different representations are provided to display the model, a mesh for each part, directly imported from the 3D-printer files. As atom centres and radii are extracted from the Peppytide model, we also provide a representation with spheres for each atom. Note that linking this simulator to UnityMol [16] is straightforward and could provide various representation types.

As stated before, an OSC server in Unity listens to the connected clients and executes based on the API defined that are initially interpreted from the different hardware events.

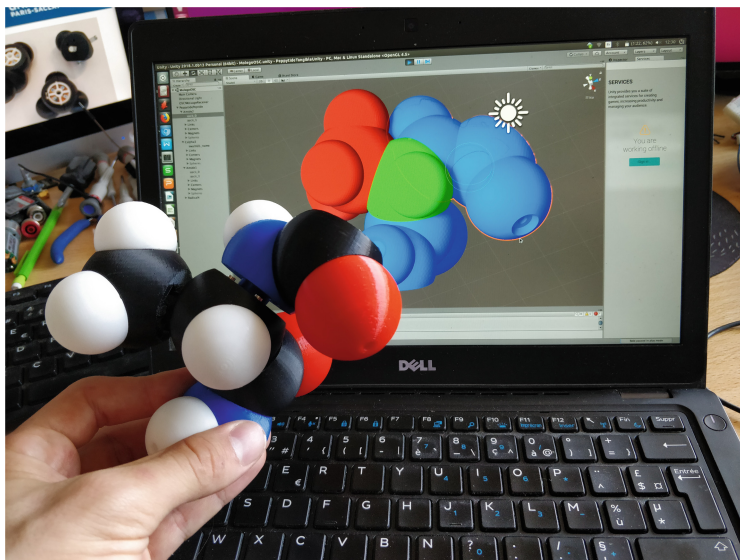


Fig. 7. Assembled molecules and their rendered Unity avatar

The binding of parts is managed programmatically: when a “connect” command is called, the non-existing parts are created and bound using a hinge joint

to allow rotation between parts but still fixing the distance between them. Angles between parts are also set using commands sent by the hardware at a desired frequency. For the first version of the simulator, the parts are not moved by adding forces to the physic simulation but by directly setting the rotation of each part. While we did not observe any artefact because the physical model parameters closely match the numerical model, this could cause issues when setting an angle between two parts that implies collision in the physic simulation. Indeed, sphere colliders are attached to each atom to reproduce steric clutter and match physical model behaviour.

One limitation of the current system is to forbid cyclic molecules but will be fixed in a future version.

In Fig. 7, we show an example of a model with its rendered avatar in Unity.

4 Evaluation

4.1 Sensors

The protein is built incrementally. Indeed, the molecule parts are placed one after the other according to the angles measured. The angle measurement error therefore implies an error in the positioning of each molecule. It is important to quantify this error. We have calculated the uncertainty on the angle measurement to determine its accuracy. After taking 50 independent measurements of an angle, we calculated an uncertainty of $\pm 0.07^\circ$. We also notice a measured noise on the angular measurement in the range of $\pm 0.2^\circ$. Considering a 50 cm protein, composed of 10 modules, one would obtain a maximum of positioning error less than 2 cm end to end, which is acceptable. In addition, we consider the possibility of using the position of potential peaks to correct these errors.

4.2 Wireless Communication

The wireless network is fully automated. All it requires is to assemble at least two elementary compounds to display them in the simulator. From that, each added compounds will be rendered.

All of this is enabled with a minimal latency for each node. Tests with up to 40 sensor updates per second were conducted and the network was stable with no failure. This sampling rate is 10 times higher than the minimum required to run an interactive model-simulation IoT interface with the protein chains. This means the human operator will not feel any lag or latency while operating the model they chose to build.

5 Future Work and Conclusion

We have instrumented a physical molecular model to be able to reconstruct its structure in real time on a simulator. All the modules are functional and we

were able to reconstruct a complete amino-acid (Amide/Amide/CarbonAlpha/Methyl). However, we still need to carry out tests on the reconstruction of bigger molecules. These large reconstructions will certainly require the use of an additional external sensor. Indeed, the main limitation of our system is the impossibility to measure the absolute orientation of the model. We only measure the relative orientation of one molecule part compared to another. The use of a conventional inertial unit is impossible because of the use of magnets with strong magnetic fields. We are therefore considering coupling our model and a camera tracking system.

References

1. Gillet, A., Sanner, M., Stoffer, D., Olson, A.: Tangible augmented interfaces for structural molecular biology. *IEEE Comput. Graphics Appl.* **25**(2), 13–17 (2005)
2. Gillet, A., Sanner, M., Stoffer, D., Olson, A.: Tangible interfaces for structural molecular biology. *Structure* **13**(3), 483–491 (2005)
3. Chakraborty, P., Zuckermann, R.N.: Coarse-grained, foldable, physical model of the polypeptide chain. *Proc. Natl. Acad. Sci.* **110**(33), 13368–13373 (2013)
4. Martinez, X., Férey, N., Vézien, J.-M., Bourdot, P.: 3D reconstruction with a markerless tracking method of flexible and modular molecular physical models: towards tangible interfaces. *J. Virtual Reality Broadcast.* **14**(2) (2019)
5. Jacobson, A., Panozzo, D., Glauser, O., Pradalier, C., Hilliges, O., Sorkine-Hornung, O.: Tangible and modular input device for character articulation. *ACM Trans. Graph. (TOG)* **33**(4), 82 (2014)
6. Atmel. ATMega328p Datasheet, January 2015. <https://www.sparkfun.com/datasheets/Components/SMD/ATMega328.pdf>
7. Nordic Semiconductor. ATMega328p Datasheet, March 2008. https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Pluss_Preliminary_Product_Specification_v1.0.pdf
8. Atmel. Bootloader. <https://www.arduino.cc/en/Hacking/Bootloader?from=Tutorial.Bootloader>
9. TMHr20. Optimized High Speed Driver for nRF24L01(+) 2.4G Hz Wireless Transceiver, March 2014. <http://tmrh20.github.io/RF24/>
10. Atmel. SPI library. <https://www.arduino.cc/en/Reference/SPI>
11. MySensors Team. MySensors: home automation and Internet of Things, 23 March 2014. <https://www.mysensors.org/>
12. Millervet. NRF24L01+ Multiceiver Network (2014). <https://www.instructables.com/id/NRF24L01-Multiceiver-Network/>
13. Dejan Nedelkovski. NRF24L01+ Tree Network, September 2018. https://howtomechatronics.com/author/howtom12_wp/
14. MySensors Team. MYSbootloader, 1 February 2015. <https://github.com/mysensors/MySensorsBootloaderRF24>
15. MySensors Team. FOTA (Wireless Programming): Firmware Over the Air (2016). <https://www.mysensors.org/about/fota>
16. Lv, Z., Tek, A., Da Silva, F., Empereur-Mot, C., Chavent, M., Baaden, M.: Game on, science-how video game technology may help biologists tackle visualization challenges. *PLoS ONE* **8**(3), e57990 (2013)