OpenMRH: a Modular Robotic Hand Model Generator Plugin for *OpenRAVE*

F. Sanfilippo [1] and K. Y. Pettersen [2]

[1] Department of Maritime Technology and Operations, Aalesund University College, Postboks 1517, 6025 Aalesund, Norway,
fisa@hials.no

[2] Department of Engineering Cybernetics, Norwegian University of Science and Technology, 7491 Trondheim, Norway,
kristin.y.pettersen@itk.ntnu.no

11^{th} International Conference on Robotics and Biomimetics (ROBIO 2015), Zhuhai, China

Summary

Bio-inspired robotic hands

Mimicking the human hand's ability, one of the most challenging problem in bio-inspired robotics:

- large gap in terms of performances.

Classical approach, analysis of the kinematic behavior of the human hand:

- simplified human hand models with minimum and optimal degrees of freedom[1], efficient manipulation tasks. Difficult to adapt to different grasping operations or to the grasping of objects with dissimilar size.

Modular grasping, a promising solution:

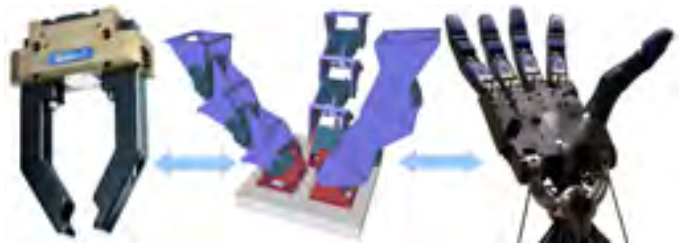
- minimum number of degrees of freedom necessary to accomplish the desired task.

[1] S. Cobos, M. Ferre, and R. Aracil. "Simplified human hand models based on grasping analysis". In: *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2010, pp. 610–615.

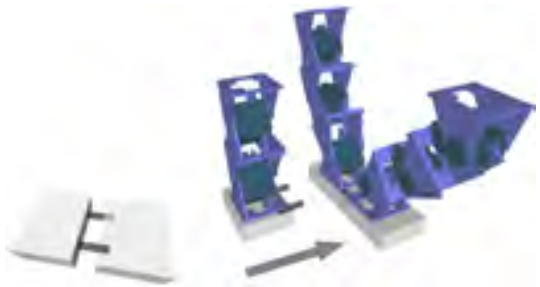
Modular grasping

Modular grasping:

- identical modules are used to build linkages in order to realize the grasping functions. From a mechanical point of view, even if it is not the most efficient grasping approach, the modular grasping still meets the requirements of standardization, modularisation, extendibility and low cost.



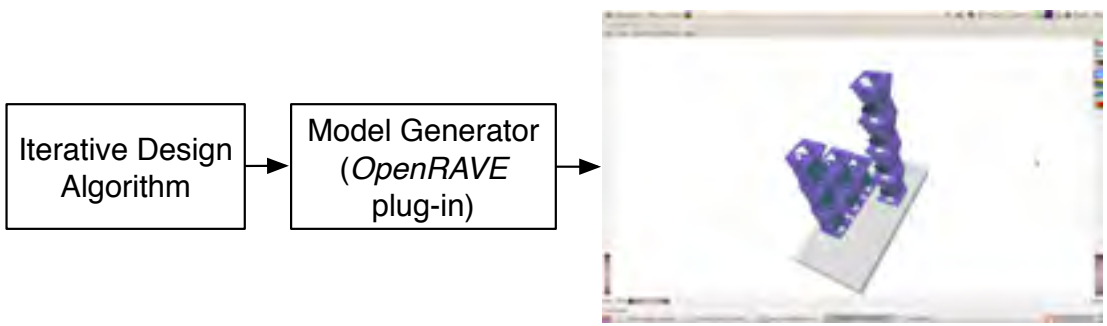
- A trade off between a simple gripper and more complex human like manipulators.
- *Principle of minimalism*: choose the simplest mechanical structure, the minimum number of actuators, the simplest set of sensors, etc.



OpenMRH: a modular robotic hand model generator plugin for *OpenRAVE*

An efficient modular grasping design method:

- A novel modular robotic grasping algorithm that allows for finding a trade-off between a simple gripper and more complex human like manipulators was previously introduced by our research group[2].
- However, all the different configurations were initially manually built by using the *Open Robotics Automation Virtual Environment* (*OpenRAVE*), a simulation environment that allows for testing, developing, and deploying motion planning algorithms. The manual process of generating different modular configurations required a relevant effort for the designer in terms of time.



OpenRAVE

[2] Filippo Sanfilippo et al. "Efficient modular grasping: an iterative approach". In: *Proc. of the 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob), Rome, Italy*. 2012, pp. 1281–1286.

A generalised manipulator model

- The simplest mechanical structure was chosen, with the minimum number of actuators, simplest set of sensors, etc. These guidelines brought forth the *Y1* modular robot[3].
- The generalised modular model consists of one or more kinematic chains of modules fixed to a base, in which each module is a chain link. When compared to a human hand, each chain represents a finger, each module represents a phalanx and the base represents the palm. The base of the model is also modular.
- Three possible base configurations are defined: *linear base* when there is no finger opposition, *circular base* when equidistant fingers are set in a circular configuration and *opposable-fingers base* when one or more fingers oppose the others.

[3] J. Gonzalez-Gomez et al. "Locomotion capabilities of a modular robot with eight pitch-yaw-connecting modules". In: *Proceeding of CLAWAR*. Citeseer. 2006, pp. 12–14.

Modular grasping design algorithm

$m(i)$	Current num. of modules
M	Maximum num. of mod. per finger
$f_{min}(i)$	Minimum num. of fingers
$Q_{desired}$	Predefined desired grasp quality

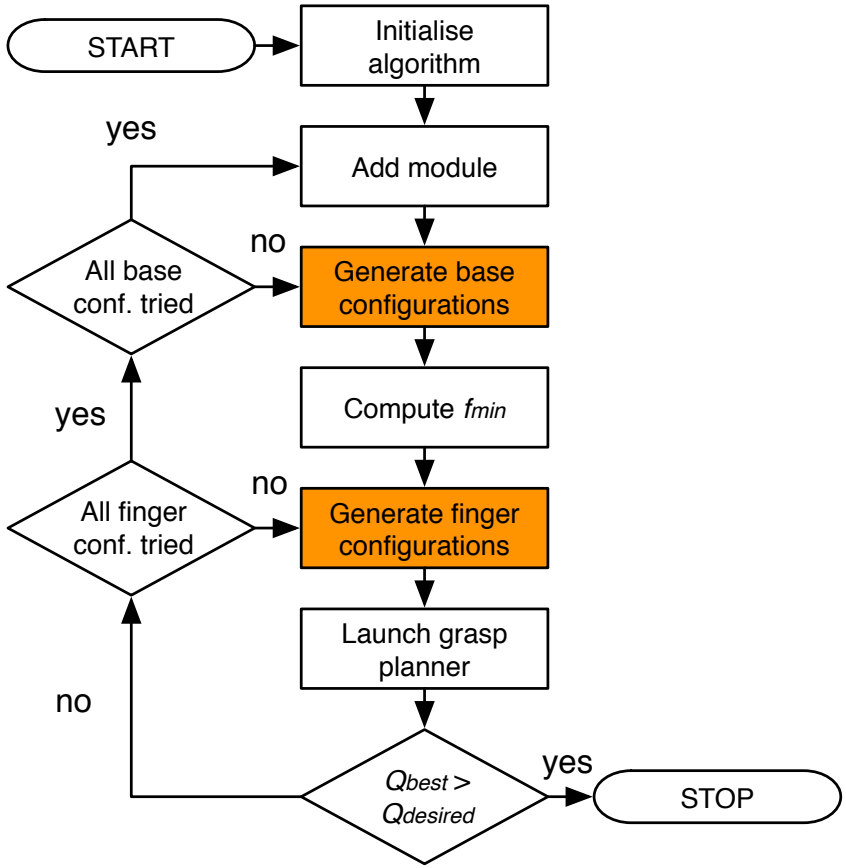
$$M_{min} = \left\lceil \frac{R}{L} \right\rceil, M_{max} = \left\lfloor \frac{2\tau_{max}}{Lw} \right\rfloor. \quad (1)$$

Compute f_{min}.

$$\frac{m(i)}{f_{min}(i)} \leqslant M. \quad (2)$$

If the inequality is not satisfied, f_{min} is incremented by one to avoid the insertion of more than M modules into a finger.

Launch planner. A grasp planner is used in order to determine the grasps achievable with the current configuration (forward solution implemented in "Openrave").



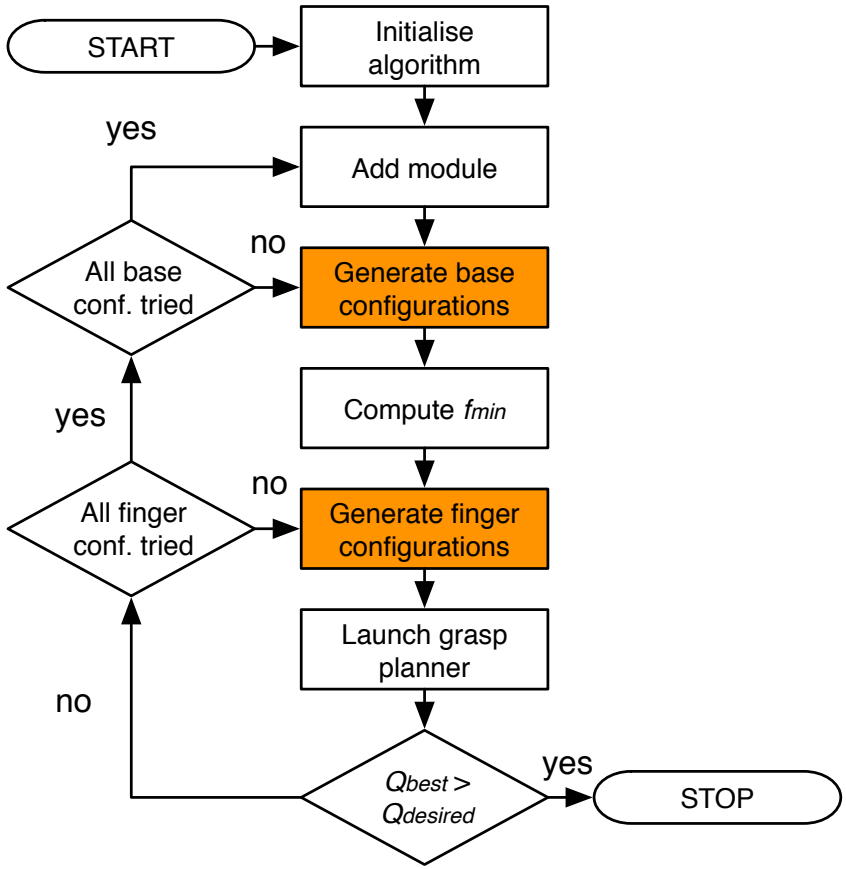
Modular grasping design algorithm

Evaluate grasp. Quality criteria introduced by Ferrari and Canny.

In particular we use the radius of the largest inscribed sphere centered at the origin contained in the *Grasp Wrench Space* (GWS). The GWS is the set of all wrenches capable of being resisted by a grasp when unit contact forces are applied at the contact points.

$$GWS = ConvexHull\left(\bigcup_{i=1}^{n} \{w_{i,1}, \ldots, w_{i,k}\}\right)$$

However, all the other solutions could be implemented and used in our algorithm.



[4]

[4] C. Ferrari and J. Canny. "Planning optimal grasps". In: *Proc. of the IEEE International Conference on Robotics and Automation*. 1992, pp. 2290–2295. ISBN: 0818627204.

Overview of *OpenMRH*

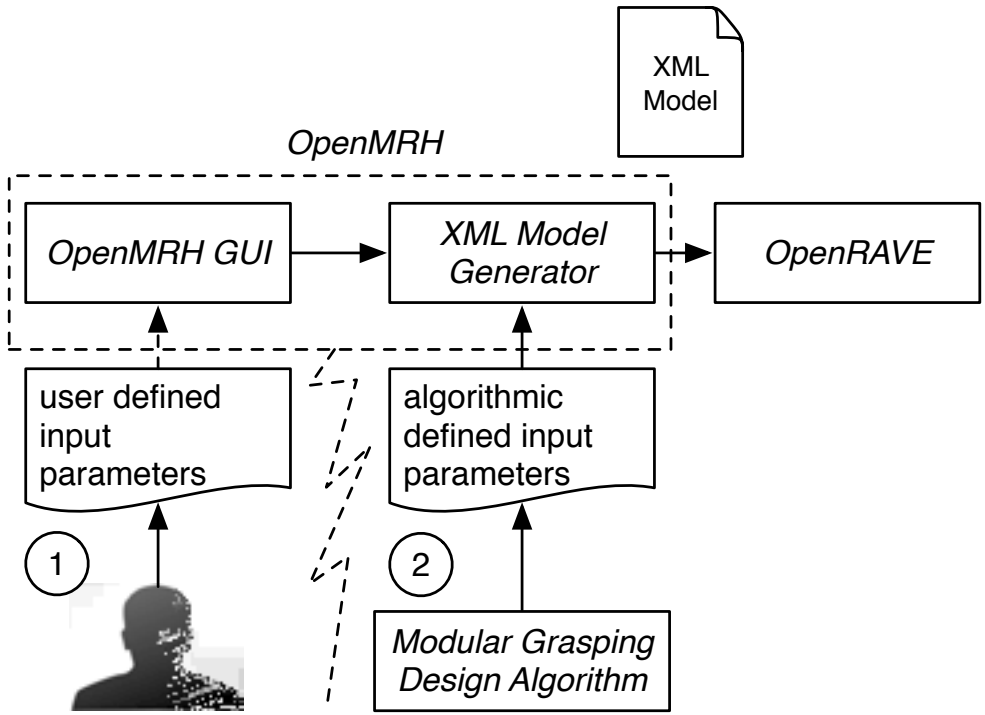
OpenRAVE[5] uses the Extensible Markup Language (XML) to store all robot and scene descriptions. A robot manipulator can be defined as a kinematic chain of its joint hierarchy. By using different XML tags, several interface types can be defined as follows:

- the *Environment* tag can be used to specify multiple robots and objects. Some GUI properties such as the camera's start location and background colour can also be defined with this tag. The *Environment* tag permits the creation of any *OpenRAVE* interface. Each interface has a type attribute to be used in specifying the interface type and defining custom XML readers;
- the *KinBody* tag can be used to define the basic object from which all other objects derive. A collection of rigid bodies and connective joints make up a kinematic body;
- the *Robot* tag is a basic robot interface that derives from the *KinBody* tag. Usually a *Robot* tag has at least one *KinBody* declaration inside it. A list of *Manipulator* and *AttachedSensor* objects can also be included within a *Robot* tag, in order to describe the robot's manipulation and sensing capabilities.

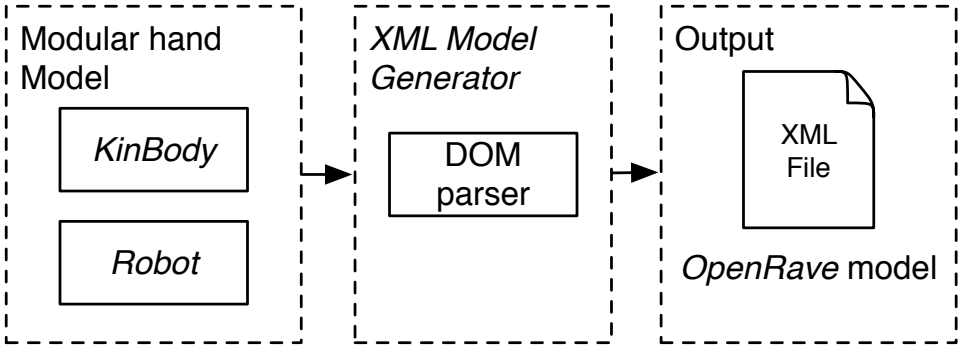
[5] Rosen Diankov and James Kuffner. "Openrave: A planning architecture for autonomous robotics". In: *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-08-34* 79 (2008).

OpenMRH architecture

- Even though this XML-based interface is quite flexible and intuitive, *OpenRAVE* does not originally provide any tools for an automated generation of robotic models.
- To overcome this issue and to improve our previously proposed design algorithm, we propose *OpenMRH*.

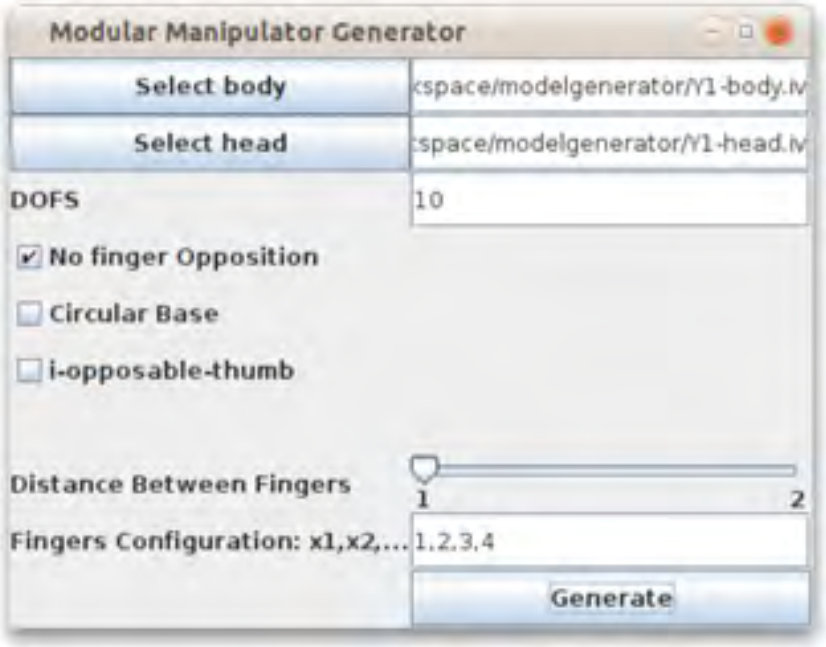


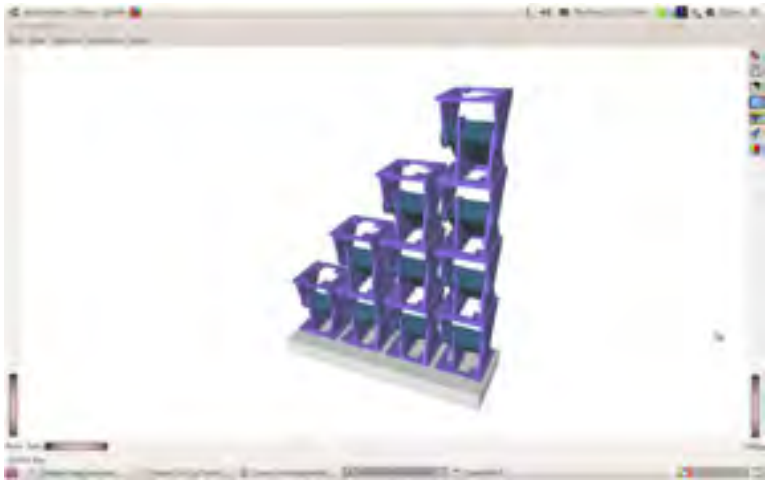
XML Model Generator



- The *XML Model Generator* makes it possible to automatically generate all the necessary XML files.
- Once the model properties are defined with either a user defined input or programmatically, the *XML Model Generator* initialises the process of creating the corresponding XML document.
- The creation of the XML document is implemented in Java by using the XML Document Object Model (DOM) parser. The XML DOM defines a standard for accessing and manipulating XML documents. The underlying idea is very simple. A DOM object with the desired tree structure is created, then the DOM object is written into a stream, in our case an XML file.
- Once the model is generated, an instance of *OpenRAVE* can be launched and the model can be visualised within the selected simulation environment.

Case study: a modular hand model generated with *OpenMRH* by using user defined input parameters





Case study: a modular robotic hand generated by using algorithmic defined input parameters

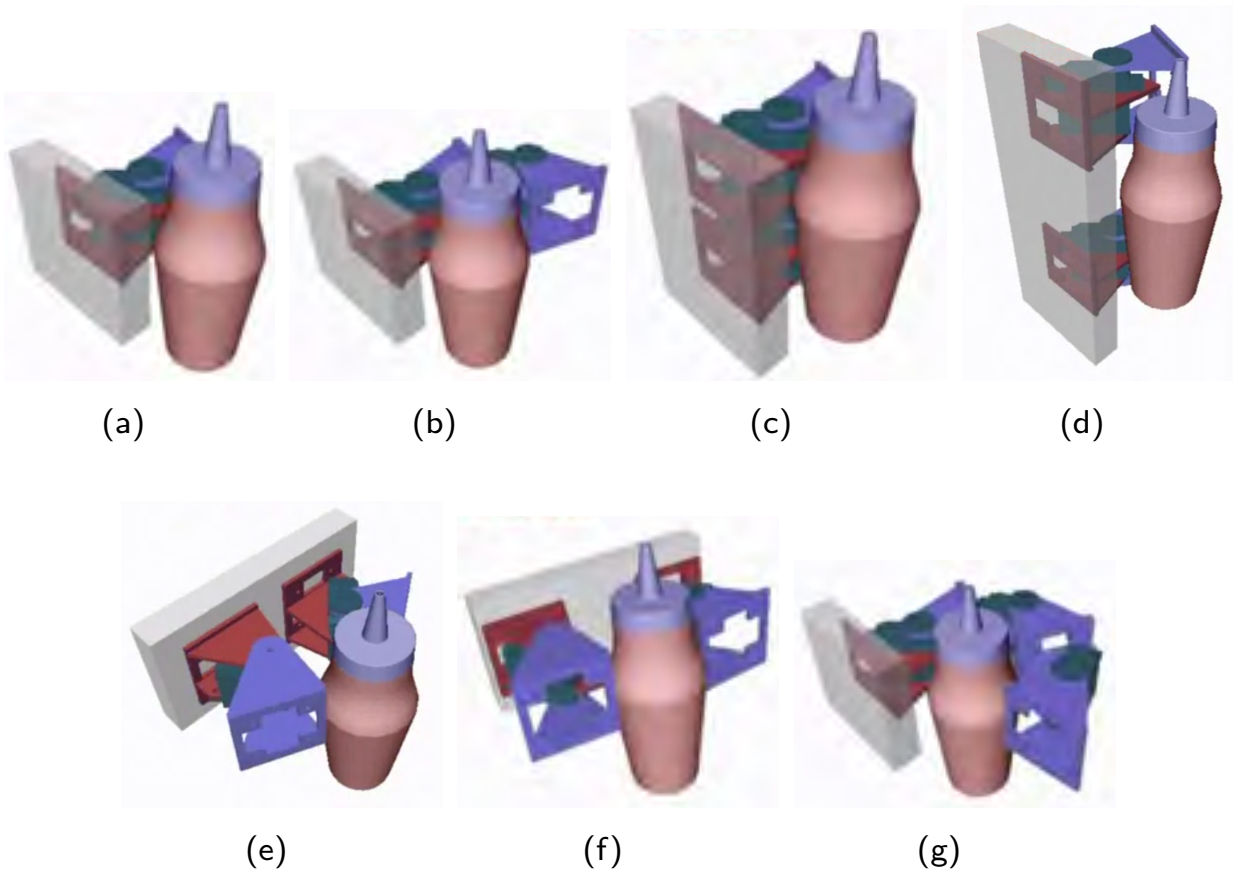


Figure: From (a) to (g), the necessary algorithm iterations to find an efficient modular configuration to grasp a common ketchup bottle. The first modular configuration able to reach the desired grasp quality is shown in (g).

Conclusion and future work

OpenMRH:

- *OpenMRH*, allows for a fast and automated generation of different modular hand models. The models can be generated either from a user defined input by using the *OpenMRH GUI* or programmatically by adopting our previously presented design algorithm.
- *OpenMRH* can be adopted for different educational and research purposes. It represents a useful extension for the *OpenRAVE* simulation environment. This extension opens up to a variety of possible application scenarios, making it possible to develop alternative design approaches and control methods for modular robotic hands. The virtual-prototyping approach can be easily combined with the modular concept.

Future work:

- A possible future work could consider the integration of the presented system with *ModGrasp*[6–8], an open-source virtual and physical rapid-prototyping framework developed by our research group.

[6] Filippo Sanfilippo et al. "ModGrasp: an Open-Source Rapid-Prototyping Framework for Designing Low-Cost Sensorised Modular Hands". In: *Proc. of the 5th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob), São Paulo, Brazil*. 2014, pp. 951–957.

[7] Filippo Sanfilippo, Houxiang Zhang, and Kristin Ytterstad Pettersen. "The New Architecture of ModGrasp for Mind-Controlled Low-Cost Sensorised Modular Hands". In: *Proc. of the 2015 IEEE International Conference on Industrial Technology (ICIT2015), Seville, Spain*. 2015, pp. 524–529.

Thank you for your attention

OpenMRH repository and support:

- *OpenMRH* is an open-source project and it is available on-line at `https://github.com/aauc-mechlab/openMRH`, along with several class diagrams and demo videos;
- F. Sanfilippo, Department of Maritime Technology and Operations, Aalesund University College, fisa@hials.no.

[1] S. Cobos, M. Ferre, and R. Aracil. "Simplified human hand models based on grasping analysis". In: *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2010, pp. 610–615.

[2] Filippo Sanfilippo et al. "Efficient modular grasping: an iterative approach". In: *Proc. of the 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob), Rome, Italy*. 2012, pp. 1281–1286.

[3] J. Gonzalez-Gomez et al. "Locomotion capabilities of a modular robot with eight pitch-yaw-connecting modules". In: *Proceeding of CLAWAR*. Citeseer. 2006, pp. 12–14.

[4] C. Ferrari and J. Canny. "Planning optimal grasps". In: *Proc. of the IEEE International Conference on Robotics and Automation*. 1992, pp. 2290–2295. ISBN: 0818627204.

[5] Rosen Diankov and James Kuffner. "Openrave: A planning architecture for autonomous robotics". In: *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-08-34* 79 (2008).

[6] Filippo Sanfilippo et al. "ModGrasp: an Open-Source Rapid-Prototyping Framework for Designing Low-Cost Sensorised Modular Hands". In: *Proc. of the 5th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob), São Paulo, Brazil*. 2014, pp. 951–957.

[7] Filippo Sanfilippo, Houxiang Zhang, and Kristin Ytterstad Pettersen. "The New Architecture of ModGrasp for Mind-Controlled Low-Cost Sensorised Modular Hands". In: *Proc. of the 2015 IEEE International Conference on Industrial Technology (ICIT2015), Seville, Spain*. 2015, pp. 524–529.

[8] Filippo Sanfilippo. "Alternative and Flexible Control Approaches for Robotic Manipulators: on the Challenge of Developing a Flexible Control Architecture that Allows for Controlling Different Manipulators". PhD thesis. Trondheim: Norwegian University of Science, Technology, Faculty of Information Technology, Mathematics, and Electrical Engineering, Department of Engineering Cybernetics, 2015.