HAL Robotics Framework

Tristan Gobin[1,2,3] **Sebastian Andraos**[1] **Thibault Schwartz**[1] **and Rémi Vriet**[1]

[1]HAL Robotics, London, United Kingdom
[2]Laboratoire GSA, Ecole Nationale Supérieure d'Architecture de Paris-Malaquais, France
[3]Laboratoire Navier, UMR 8205, Ecole des Ponts, CNRS, UGE, Champs-sur-Marne, France

t.gobin@hal-robtics.com, the-team@hal-robtics.com

Abstract -

The latest technological developments, especially in software, have made it possible to lower the barrier to entry for robotics, notably in fields that have typically been under-automated, like construction. Robotics in the construction industry is not new, but its acceleration has been marked in the last 10 years. This article presents the latest evolution of HAL Robotics' software, now called the *HAL Robotics Framework* alongside its associated concepts, its technical features, and its use in manufacturing and construction.

Keywords -

Framework; Robotics; Programming; CAD; Onsite; Offsite; Software; Simulation;

1 Introduction

Initially, the software was an academic initiative to simplify the programming of industrial robots for use in the teaching and research of architecture [1]. It was then developed and enhanced alongside several research projects primarily focused on the fabrication of complex elements, including concrete lattices and moulds [2], steam-bent timber [3] and hot-wire-cut foam for stereotomy studies [4], before being rewritten from the ground up from 2015 onwards to bridge the gap between R&D processes and industrialization.

This invariably led to several questions concerning the automatization of construction processes. First and foremost, what is required to have a seamless flow of data from design to fabrication? What kind of machines and mechanisms are sufficiently adaptable to various manufacturing processes, accurate and affordable enough to be appropriate for construction? And, given that robotics and programming are unlikely to be in the inherent skill-sets of architects and engineers, how can construction experts propose and experiment with new processes? With well proven economic and distribution models, easy adaptability for many processes, and a modular approach to their installation, six-axis arms, like those found in automotive factories, seemed to be a good starting point. Although flexible, these machines have constraints which must be presented to, and understood, by their users alongside tools to simplify the creation of paths, simulation and communication, amongst others.

To address these requirements the *HAL Robotics Framework* is structured around four main features:

a modelling: assembly of components required to virtually represent a robot cell.

b programming: vendor-agnostic definition of where a machine should move, pause and trigger any events.

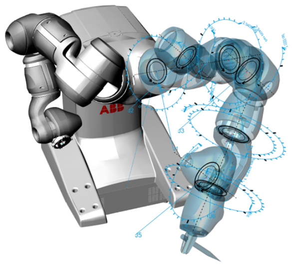
c simulating: high-fidelity motion planning for one or more virtual machines to validate programs.

d communicating: data exchange with physical, networked devices, be they robot programs being uploaded or sensor values monitored in real-time.

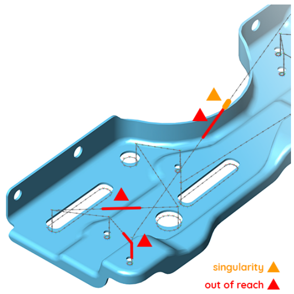
1.1 Modelling

Before programming any process a virtual representation of the robotic cell must be modelled. Typically (as shown in Figure 2) this cell model includes passive elements, such as fencing or other contextual items surrounding the robots, mechanisms that are controlled, such as robotic manipulators and positioners, process equipment, such as end-effectors or sensors, and controllers and PLCs administering all the equipment that will be programmed. We have developed software layers to help build these digital twins from the instantiation of virtual mechanisms and their controllers to the assembly of complex, bespoke, mechanical systems including the integration of these tools into the CAD environments and catalogues of prebuilt models that can be simply dropped in. These catalogues also offers basic research tools and allow sorting prebuilt mechanisms by payload, reach, names, or other criteria. This allows users to easily swap robots to find the best fit during cell specification, change or evolve processes for a new cell or test different robot brands to fit their needs.

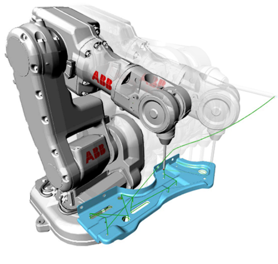
The most complex catalogue is that of the robot controllers, each instance of which is configurable to match the options, e.g. communication protocols or language



(a) Model



(b) Program



(c) Simulate

(d) Communicate

Figure 1. Key feature categories of the *HAL Robotics Framework*

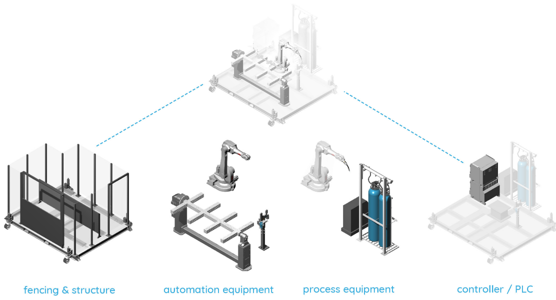


Figure 2. Core components of a model can be instantiated from catalogues of prebuilt items or manually modelled to create a digital representation of a cell ready for programming.

support, that are available on its physical counterpart and may be used to connect directly to it over the network. Several brands' controllers are currently supported, each of which propose subtly different implementations of similar capabilities. These option implementations are presented to users as *controller subsystems* which can be swapped to change how, among other things, a controller exports programs, uploads them to a remote controller (including authorization credentials and network details), starts or stops a running mechanism, monitors a mechanisms joint positions or I/O signals etc. As well as allowing a faithful representation of the equipment a user has at their disposal, this modular approach allows controllers to be extended and have new features added alongside manufacturers' updates without breaking existing uses or assuming that all users will invest in these latest packages. The graphical user interface (GUI) for configuring controllers from different manufacturers can be seen in Figure 3).

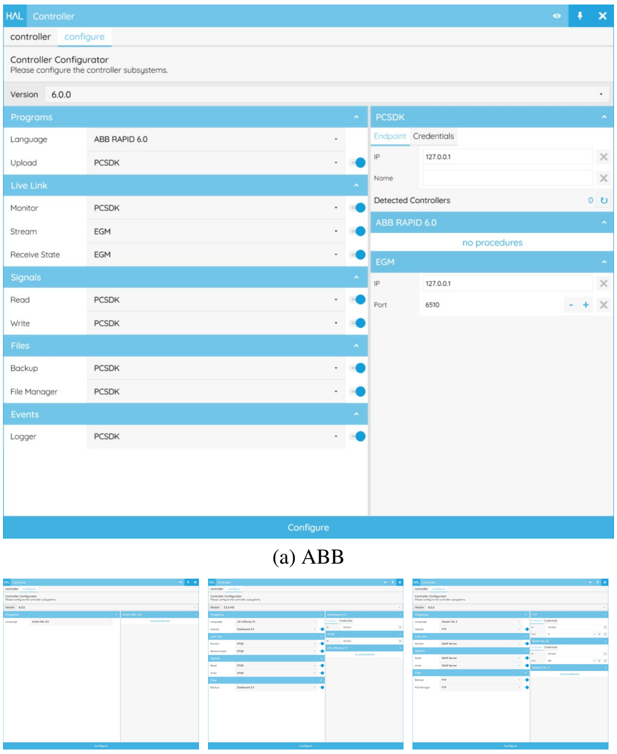
1.2 Program and Simulate

The prototypical workflow using the *HAL Robotics Framework* (shown in Figure 5) is built around linking process automation to CAD-based part descriptions and geometry. This approach, notably when the *HAL Robotics Framework* is integrated into CAD applications, ensures that trajectories are associative, that is to say that when the part is modified or moved, the trajectory will automatically adapt itself to match new revisions.

This flexible programming of toolpaths is built around *Actions*, the fundamental things a robot can do such as moving to a certain target, changing signal states or communicating with other devices like sensors, being parametrically linked to part geometry and subscribed to changes.

A sequence of *Actions* are combined into a *Procedure*. A *Procedure* can be "solved" by computing not only where a mechanism will move but *how* it will do so taking into account user-specified constraints like blends and inherent constraints like the maximum speeds and accelerations of each robot joint. From this solved state, we can run diagnoses, typically reachability of targets, collision checks, and kinematic singularities, any instances of which can be visualised help the user understand how to fix their issues (see Figure 7). These will help inform the user to optimise a robot's motion settings or even redesign a part if it cannot be processed with the given constraints, or could be fabricated more efficiently with minor modifications. This roughly follows the principles of Robot Oriented Design proposed by T. Bock [5] and expanded in Linner and Bock [6] by giving users, within their CAD environment, immediate graphical insights into how their design choices are impacting, and are impacted by, process and robot constraints. One of the major ramifications of this associative and parametric programming style is that many variants of a part can be swapped into the same *Procedure* and the robot programming is almost instantaneously updated, reducing the cost of programming additional variable parts close to nil and enabling mass customisation.

An important aspect at this stage is that all *Procedures* are "vendor-agnostic", meaning that they are not linked to any given manufacturer's language and can be translated



(a) ABB

(b) KUKA (c) Universal Robots (d) Staubli

Figure 3. Controller subsystem configurator which allows users to change how capabilities are implemented in their virtual controller and how it may communicate with the real.

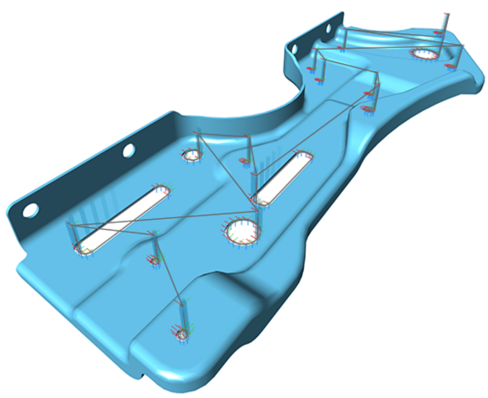


Figure 4. Procedures are associated to part geometry such that when, in this example, a hole is moved or modified, the toolpath will automatically adapt itself to the changes.

for any machine, making it possible to revise the design of a production cell, such as changing the robot(s), adding a positioner, changing the dimensions of the tool etc., all while still working on the part design.

1.3 Procedure Analysis and Debugging

Controlling a robotic manipulator, or any other mechanism, involves determining a set of appropriate joint positions such that the end effector(s) are moved to desired positions (also called "inverse kinematics") as smoothly, rapidly, and accurately as possible. This is generally termed "motion planning". To solve inverse kinematic problems, the *HAL Robotics framework* uses robust numerical approaches such as the *jacobian pseudoinverse* [7], *damped least squares* with *singular value decomposition* (SVD-DLS) and *selectively damped least squares* (SDLS) [8].

These solutions are suitable for most complex mechanisms and offer real-time, adaptive control, thereby enabling the use of additional motion constraints to achieve secondary goals such as avoiding of joint limits, singularities or obstacles. The accurate modelling of robot kinematics provides users with advanced information about their real robot's behaviour. These details, such as speed, acceleration, position or torque of each joint, can be used to analyse motion behaviour and get a deeper understanding of mechanism's motion. For example, in a 3d printing process, one could examine the end effector speed to ensure it moves at a constant speed or determine that one joint's maximum speed is reached intermittently limiting end effector speed and leading to inconcsitent material deposition.

A GUI dashboard and timeline, have been developed to easily represent the chronology of *Procedures*, which can often become complex particularly when dealing with multiple machines in parallel and constantly shifting between synchronous (robots moving at the same time) and asynchronous (robots moving separately or waiting for each other). The dashboard, shown in 7, also display errors and warnings for each *Procedure* and allows the setting of "breakpoints" at which the simulation will automatically stop (useful for pinpointing the source of issues).

1.4 Linguistics, Communication and Interoperability

Once programmed *Procedures* have been validated in the virtual, they need to be executed on physical machines to undertake their processes. This is the stage at which our vendor-agnostic procedure gets converted into manufacturer-specific machine code, which can, in turn, be uploaded to the machine via an external medium like a USB key or, ideally, over a network. In the latter case we are directly connected to the controller and therefore, subject to support in the form of a *controller subsystem*, we can start to bring real-time data back from the machines which can be used to optimize process settings, evaluate tolerances, estimate energy costs, and/or we can store this data for traceability.

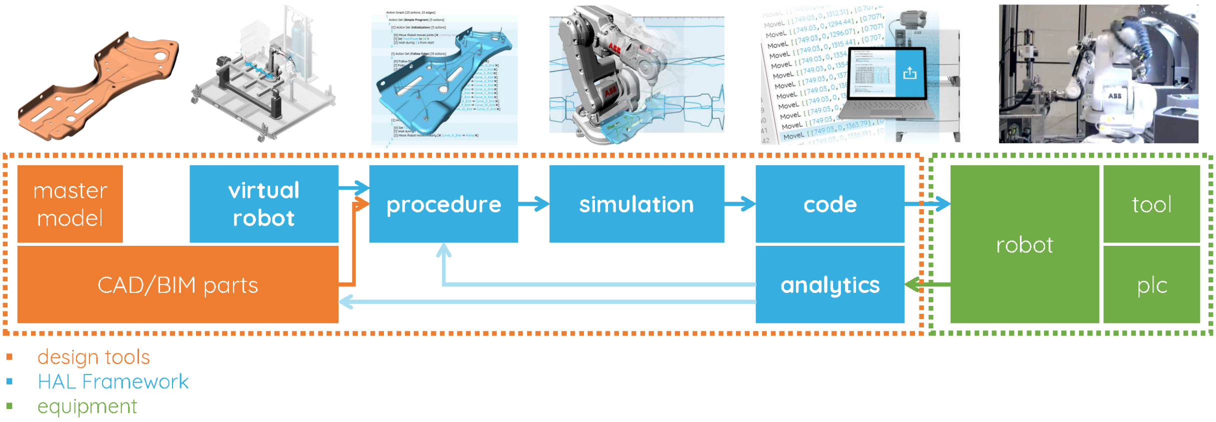


Figure 5. Typical HAL Robotics Framework workflow, from a CAD part to robotic execution and data feedback.

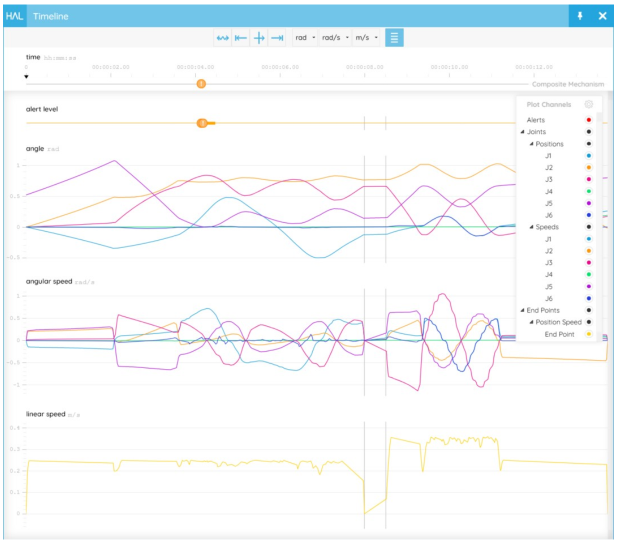


Figure 6. Timeline of a solved procedure showing the computed positions and speeds of each robot joint and the end effector.

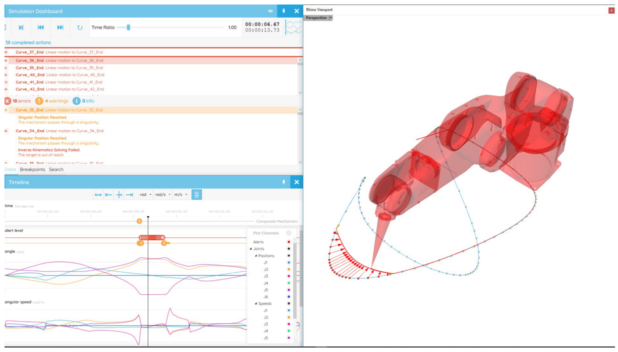


Figure 7. Dashboard showing the chronology of a Procedure alongside a robot in an error position highlighting the difference between its real and programmed toolpaths.

Adaptive Trajectories Once we can send data to a machine and read data from it then we can start to play with integrating sensors into the loop and create highly adaptive programs, combining sensed data from the real world with the digital data from the virtual world. This project at Carnegie Mellon University [3] involved steam-bending wood which, being a heterogeneous material, can't be easily simulated. An array of infra-red cameras (an OptiTrack motion capture system) and reflective markers on the wooden element calculated the current position of the real wood, compared it to the idealized digital model, and tweaked the trajectories of the two robots independently, and in real time, to create the exact shape required.

These same principles can be used to operate a machine through human motion and gestures. In [9] and its related workshop, participants used their right hand to control the motion of a robot and gestures on their left to control I/O signals and thereby open or close a gripper to manipulate parts. Hand tracking and gesture recognition was handled by a Leap Motion and robot control through a combination of ABB's Externally Guided Motion (EGM) and custom protocols.

Embedded Software Combining this notion of one data transmission with the fact that the *HAL Robotics Framework* is lightweight and compatible with everything down to low-cost Linux hardware, means that the entire feature set of the *HAL Robotics Framework*, as well as any control logic that might be required, can be embedded within a production line or even a robot controller[10].

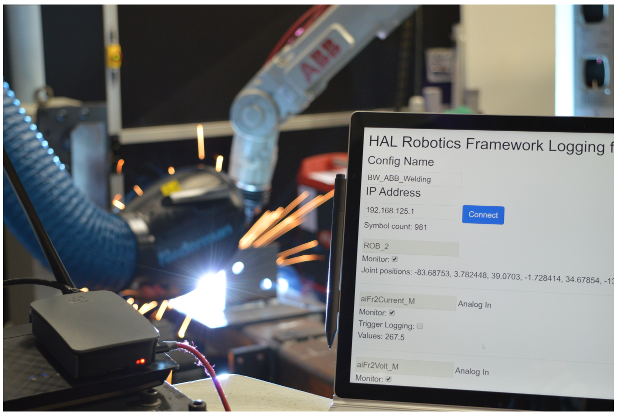


Figure 8. Monitoring welding data (joint positions, currents and voltages) through the HAL Robotics Framework running on a low-cost Raspberry Pi.

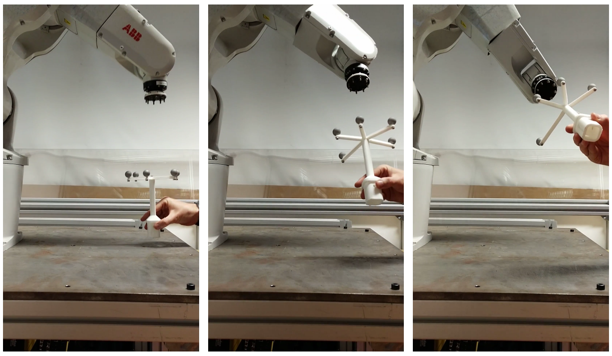


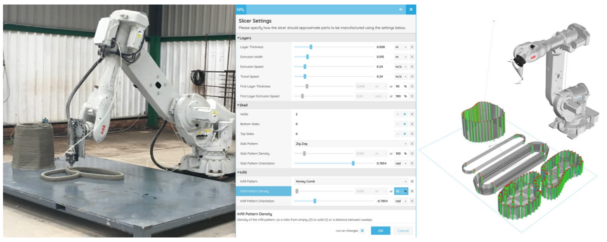
Figure 9. Robot following motion capture markers in 6 DoF using an OptiTrack camera array and ABB EGM.

2 Multi-process

The *HAL Robotics Framework* has been used by numerous academic, industrial and research and development institutions around the world to greatly reduce the development time of their novel manufacturing processes, including 3d printing concrete parts [11, 12], machining timber frames [13], prototyping plane components, non-destructive testing (NDT) [14] and even decorating cakes. Catering to, and receiving feedback from, this broad spectrum of processes ensured that the *HAL Robotics Framework* could handle, not only many different processes, but also multiple processes during the same *Procedure* (multi-process), greatly expanding the flexibility of these machines.

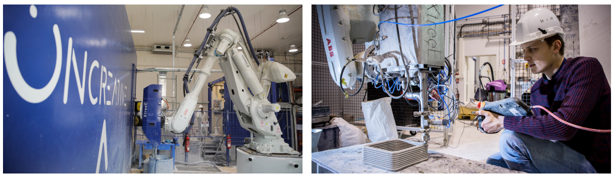
To approach multi-process programming, we start from the principle that we can use machines without knowing in advance what the users will do with them forcing the generalisation of robotic capabilities to primitives such as motion, I/O changes and waiting. This approach clearly lacks the specificity, and simplicity, of process-dedicated software but lays a solid foundation upon which process-specific extensions can be added. These extensions, such as the slicer shown in 10, emebed the semantics of process-dedicated users and variable user interfaces to cater for different skill levels in the field, reintroducing the simplicity lost by generalisation.

3D Concrete Printing 3D printing has be touted as a game-changing process but the toolpath programming and motion control required to successfully implement it, especially with robotic systems, remain obstacles to proliferation.



(a) Slicer and 3d printing

Figure 10. A 3D printing-specific extension to the HAL Robotics Framework simplifies toolpath generation for this complex process.



(a) Concreative: industrial robotic 3DCP cell (b) ENPC: R&D robotic 3DCP cell

Figure 11. Commercial and experimental 3D printing of concrete programmed using the HAL Robotics Framework.

HAL Robotics' software and solutions have been used to underpin groundbreaking R&D from the Democrite [11] project directly printing cementitious materials, to printing moulds in clays into which concrete can later be poured [15], all the way to enabling the creation of entirely new research departments like the Build'In co-innovation lab in Paris hosted by the École des Ponts, ParisTech, and, as 3D Concrete Printing (3DCP) has matured, so too have our solutions moving from powering pure R&D to underpinning industrial startups like XtreeE, and industrial production like Concreative in Dubai.

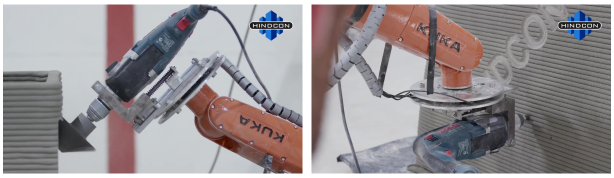
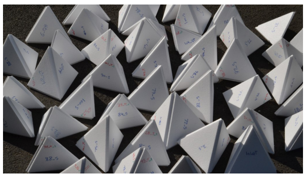


Figure 12. Hindcon: milling freshly 3D printed mortar to improve tolerances change the finish.

Other interesting projects, like Hindcon or the 3D printed Truss developed at Navier's laboratory, demonstrate how 3D printing can be enhanced by combining it with other processes. In Hindcon a milling head is used in a post-printing step to change the surface finish of freshly printed mortar. For the spatial truss, on the other hand, the robot is first used to hot-wire cut (HWC) foam blocks which are then used as a formwork for the deposition of cementitious mortar by the same robot with a printing head now mounted. This process is explained in detail in [12].



(a) HWC EPS blocks.

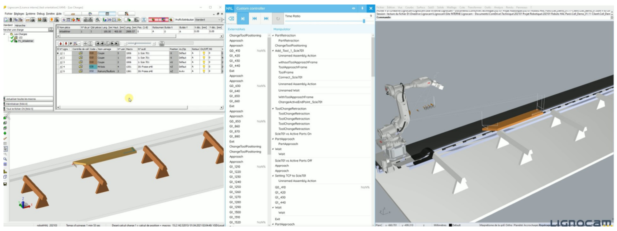
(b) Free deposition on EPS blocks.

(c) The completed truss, after removal of some EPS blocks (photo: Stefano Borghi).

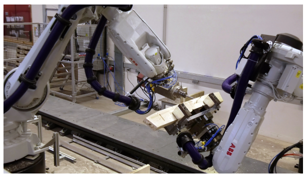
Figure 13. A 3DCP spatial truss is built up by extruding onto HWC foam blocks.

Through collaboration with industrial partners, standardised tools can be tuned to meet the exact requirements of their process or operators. In 10 we see a slicer, derived from HAL Robotics' standard slicer which handles vase mode, infill patterns, concentric walls and most other industry standard features, tailored to the requirements of their engineers and operators leaving a reduced set of parameters for their operators to use and some custom algorithms behind the scenes to fill in the gaps. This slicer retains the fully parametric approach to toolpath generation and the ability to take robotic constraints into consideration during the slicing, retaining rich metadata about the process for easy manipulation of the results, from the standard slicer.

Woodworking Through collaboration with LignoCAM, experts in woodworking software, HAL Robotics have had the opportunity to develop tools in wood machining for the manufacture of structural and architectural elements. To ensure ease of use for experts in the field, this included interoperability between the robotics, including all of their constraints, and trade-specific operations, CAD environments and standard formats (like BTL) to which carpenters are accustomed. Combining this with experience testing innovative processes, such as those carried out with the Navier laboratory [13], the know-how of this sector, and its robust ecosystem of tools, can be integrated into the *HAL Robotics Framework*.



(a) LignoCAM to HAL: BTL interpretation



(b) multi robot milling

(c) result

Figure 14. Workflow of timber manufacturing from LignoCAM to BTL through the HAL Robotics Framework to robots and the assembled product.

On-site Robotics Moving from factories to construction sites brings a new set of opportunities and challenges which have been explored through projects like COSCR [16] which included work on:

- Integration into CAD/BIM software, allowing users to design and optimize work with integrated manufacturing constraints and to generate machine code directly from their model.
- Mobile platform enabling autonomous movement and access to height, via a hydraulic mast, between

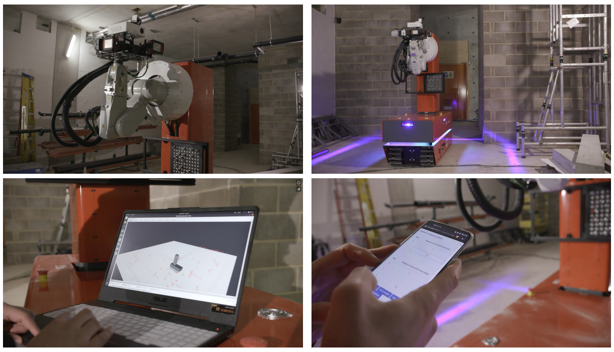


Figure 15. COSCR can autonomously navigate construction sites and execute tasks as well as being controlled manually using various intuitive input interfaces.

tasks with sufficient accuracy to carry out drilling tasks.

- Automated simultaneous localization and mapping of the construction site to enable autonomous navigation and reuse of those sensors to detect humans. Maps from measurements via the moving robot, correlated with CAD drawings, determine where tasks need to be carried out.
- Ensuring safe human/robot interaction.
- Access to multiple tools increasing flexibility of the platform.

3 Conclusion: Towards Robot Task Orchestration

With the ability to run or connect to multiple cells or machines simultaneously, questions naturally arise about how best to distribute work to these cells automatically. This problem, referred to as "orchestration" in computing, could be distributing work between similar robots on the same construction site based on proximity and availability or between different factories around the world based on logistics and certifications. An orchestrator would optimize the utilization of multiple cells to minimize downtime, increase time in use, adapt to mechanical failures and improve the overall efficiency of multi-cell factories. HAL Robotics are working with Konica Minolta to extend their Platform as a Service (PaaS) orchestrator DCI to translate the advantages it provides for computational resources to mechanical devices. This means that robots can offload computation to more powerful devices, can communicate their status to a central orchestrator, and any faults can be compensated by other robots with the same capabilities. The classification of capabilities is not a metric that is included in any existing container orchestrators and can be extremely varied including process-specific parameters e.g drilling with certain bit sizes, end shapes and compatible materials, or imaging with specific colour spaces, resolutions and exposure settings. This is further complexified as each capability translates into one or more applications that can be undertaken e.g. 2D imaging could be used for inspection, QA, and object detection. This collaboration will, therefore, create a new, extensible orchestrator that is able to incorporate the complexities and varieties of robotic capabilities for task distribution.

HAL Robotics have supported the development of many processes and projects over the past several years and will continue to do so by refining the tools made available to users and accelerate the deployment of R&D into industrial applications with packaged extensions to the *HAL Robotics Framework*. HAL Robotics will use their experience in manufacturing and construction to help advance both on and off site construction and fabrication from design all the way through to installation. The complexities and variety of requirements in construction make it the perfect sector for experimenting with advanced robotics and putting our developments to the test.

References

[1] Thibault Schwartz. Hal Extension of a visual programming language to support teaching and research on robotics applied to construction. *Rob| Arch 2012*, pages 92–101, 2013. doi:10.1007/978-3-7091-1465-0_8. URL `http://dx.doi.org/10.1007/978-3-7091-1465-0{_}8`.

[2] Philippe Morel and Thibault Schwartz. Modelling Behaviour. *Modelling Behaviour*, pages 213–223, 2015. doi:10.1007/978-3-319-24208-8.

[3] Thibault Schwartz, Joshua Bard, Madeline Ganon, Zack Jacobson-Weaver, Michael Jeffers, and Richard Tursky. All Bent Out... Adaptive Fabrication of Bent Wood Assembles. *Robotic Fabrication in Architecture, Art and Design 2014*, pages 305–317, 2014. doi:10.1007/978-3-319-04663-1. URL `https://doi.org/10.1007/978-3-319-04663-1{_}21http://dx.doi.org/10.1007/978-3-319-04663-1{_}21`.

[4] Thibault Schwartz and Lucia Mondardini. Integrated Design and Robotized Prototyping of Abeille's Vaults. In Wes McGee and Monica de Leon, editors, *Robotic Fabrication in Architecture, Art and Design 2014*, number 2012, pages 199–209. Springer International Publishing, Cham, 2014. ISBN 978-3-319-04663-1.

doi:10.1007/978-3-319-04663-1_13. URL http://link.springer.com/10.1007/978-3-319-04663-1http://dx.doi.org/10.1007/978-3-319-04663-1{_}13.

[5] Thomas-Alexander Bock. Robot-Oriented Design. In *5th ISRC*, pages 135–144. IAARC, 1988. ISBN 9781139924146. doi:10.1017/CBO9781139924146. URL http://www.iaarc.org/publications/proceedings{_}of{_}the{_}5th{_}isarc/robotoriented{_}design.htmlhttp://ebooks.cambridge.org/ref/id/CBO9781139924146.

[6] Thomas Bock and Thomas Linner. *Robot-oriented design: Design and management tools for the deployment of automation and robotics in construction*. 2015. ISBN 9781139924146. doi:10.1017/CBO9781139924146.

[7] Charles A. Klein and Ching Hsiang Huang. Review of Pseudoinverse Control for Use with Kinematically Redundant Manipulators. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-13(2):245–250, 1983. ISSN 21682909. doi:10.1109/TSMC.1983.6313123.

[8] A. Balestrino, G. De Maria, and L. Sciavicco. Robust Control of Robotic Manipulators. *IFAC Proceedings Series*, 17(2):2435–2440, 1985. ISSN 07411146. doi:10.1016/s1474-6670(17)61347-8. URL http://dx.doi.org/10.1016/S1474-6670(17)61347-8.

[9] Thibault Schwartz, Sebastian Andraos, Jonathan Nelson, Christopher Knapp, and Bertrand Arnold. Towards On-site Collaborative Robotics. In Dagmar Reinhardt, Rob Saunders, and Jane Burry, editors, *Robotic Fabrication in architecture, art and design 2016*, volume iv, pages 388–397. Springer, Cham, 2016. ISBN 978-3-319-26378-6. doi:10.1007/978-3-319-26378-6_31. URL http://dx.doi.org/10.1007/978-3-319-26378-6{_}31.

[10] Inovo Robotics, HAL Robotics. Low Cost Modular Manipulator, 2017. URL https://gtr.ukri.org/projects?ref=103675.

[11] C. Gosselin, R. Duballet, Ph Roux, N. Gaudillière, J. Dirrenberger, and Ph Morel. Large-scale 3D printing of ultra-high performance concrete - a new processing route for architects and builders. *Materials and Design*, 100:102–109, 2016. ISSN 18734197. doi:10.1016/j.matdes.2016.03.097.

[12] Romain Duballet, Romain Mesnil, Nicolas Ducoulombier, Paul Carneau, Leo Demont, Mahan Motamedi, Olivier Baverel, Jean François Caron, and Justin Dirrenberger. Free Deposition Printing for Space Truss Structures. *RILEM Bookseries*, 28:873–882, jul 2020. ISSN 22110852. doi:10.1007/978-3-030-49916-7_85. URL https://link.springer.com/chapter/10.1007/978-3-030-49916-7{_}85.

[13] Tristan Gobin, Romain Mesnil, Cyril Douthe, Pierre Margerit, Nicolas Ducoulombier, Leo Demont, Hocine Delmi, and Jean-fran Caron. Form Finding of Nexorades Using the Translations Method. *Robotic Fabrication in Architecture, Art and Design 2018*, 2019. doi:10.1007/978-3-319-92294-2.

[14] Pierre Margerit, Tristan Gobin, Arthur Lebée, and Jean François Caron. The robotized laser doppler vibrometer: On the use of an industrial robot arm to perform 3D full-field velocity measurements. *Optics and Lasers in Engineering*, 137(July 2020), 2021. ISSN 01438166. doi:10.1016/j.optlaseng.2020.106363.

[15] P I Nterest. Clay Robotics : The Future Of Architecture Is Happening Now In A Chilterns Farm. 2014. URL https://www.theguardian.com/artanddesign/architecture-design-blog/2014/aug/08/clay-robotics-architecture-chilterns-farm.

[16] HAL Robotics, InnoTecUK, Skanska, ABB Robotics, Skyjack Inc., BRE. COSCR: Collaborative, On-Site Construction Robot, 2021. URL https://www.skanska.co.uk/about-skanska/media/press-releases/257313/Skanska-supports-development-of-new-construction-robot.