

A RAPID PROTOTYPING ENVIRONMENT FOR MOBILE REHABILITATION ROBOTICS

Ir. B.J.F. Driessen, ing. J.A. v. Woerden, Prof. Dr. G. Bolmsjö (Lund University),
Dipl.- Ing. O. Buckmann (BIBA)

TNO-TPD, PO-BOX 155, 2600 AD, Delft, The Netherlands
Email: driessen@tpd.tno.nl, tel: +31 15 2692394, fax: +31 15 2692111

Abstract: This paper describes a development environment for collaborative engineering of rehabilitation robotics devices, called RETIMO. The basis of RETIMO is models of the different components (mechanics, computer hardware, controller, human interfaces) of the mobile robot. Each component can exist in three different stages: a) simulation stage, b) virtual prototyping stage, c) real prototyping stage. RETIMO will lead to:

- faster to the market
- design cost reduction because of collaborative engineering,
- better quality because end-users are more involved

An example of the method is given by the design of a mobile base mounted manipulator.

Introduction

Designing assistive devices requires a tight co-operation between developers, end-users, and therapists during the entire development process. This applies especially for *advanced* assistive devices such as robots or mobile bases. Engineers can develop advanced control functions for a mobile robot, for example an intelligent path planning algorithm for

moving to a pre-defined position from any point in the workspace of the mobile system. However, if the end-user does not have the capability to store *new* locations and postures in the memory of the mobile robot, the functionality of the path planner from the end-user's point of view is rather poor. Seemingly, the end-user interface is in this situation responsible for a partial failure of the developed functionality. Would the end-user, however, have the possibility to test and evaluate the functionality at a very early stage, the engineer could have used the feedback for adapting the functionality in such a way that the end-user can really use it.

This paper describes a development strategy for assistive devices, enabling the integration of the end-user in the development process. After a short review of development strategies for assistive (mechatronic) devices, the structure of RETIMO is explained. Emphasis is put on the controller prototyping and the embedded system prototyping. Early results of the MobiNet [1] program are given, demonstrating the current status of the development environment. Finally developments

scheduled for the near future are indicated.

Development strategy

Rapid prototyping is a technique for analysing complex problems by using fast realisation methods for different components (e.g. mechanical parts, controllers, etc.). By having prototypes of these components, end-users can interact with either the real or the virtual system at an early stage. Rapid prototyping applies to different disciplines, for example:

- mechanical rapid prototyping, dealing with the rapid manufacturing of mechanical components directly from 3D cad-drawings [2],
- embedded system rapid prototyping, dealing with evaluating different hardware architectures using processes and virtual communications,
- controller rapid prototyping, dealing with the (semi)automatic implementation of control algorithms from simulation results,
- user-interfacing prototyping, dealing with designing optimal user interfaces for end-users.

For all disciplines, three development phases can be identified: the simulation phase, the virtual prototyping phase and the prototype realisation phase¹.

During the *simulation phase*, models of the system are created for some disciplines. Using these models (which

¹ Note that during product development, design iterations between these phases is very much required.

are not real-time), calculations which are required in a certain discipline can be carried out. Examples of these calculations are: strength analysis calculations using FEM packages, control design and tuning using Matlab or MatrixX, or computational analysis calculation e.g. using HAMLET. Consequences of basic decisions can be evaluated for all disciplines. For example, what will the mechanic structure of the system be if a three fingered gripper will be used instead of a two fingered gripper. Visualisation can show the results in a more understandable format.

During the *virtual prototyping phase*, the models are compiled to a real-time environment. The assistive device still exists only in a virtual world, but now can be simulated in real-time. Therefore, the dynamic behaviour of the system can also be visualised in real-time. This makes it possible for end-users to test and evaluate the assistive device by means of 3D visualisations. Since no parts of the system exist in reality, modifications in the structure of the device can be made with limited effort. The virtual prototyping phase is very important for incorporating end-users in the development of assistive devices.

During the *prototype realisation phase*, the system will be constructed in reality. It is also possible to combine virtual prototypes with real prototypes of the system. In this way, incremental system development is possible, reducing the

risks of product failures strongly. The results of the virtual prototyping phase (3D CAD drawings, programming code of control algorithms) are reused as much as possible. This is very much required, since the prototype realisation phase is often the most time consuming during system development and iterations in this phase are very costly.

For following the described working methodology, an environment with the following requirements is needed:

- open with respect to integration standards such as M3S,
- possibility to carry out (semi) automatic model transformations between simulation model, virtual prototyping model, and real prototype,
- 3D visualisation capabilities for visualising the system to the end user as well as the engineer,
- open with respect to hardware platforms, making it possible to test different kinds of communication buses (e.g. CAN), and different kind of real time environments (PharLap, Windows CE, VxWorks, etc).
- wide availability of debugging facilities.

State of the art

No full blown methods for development of complex systems of the described nature are found. The literature addresses single discipline methods for many applications. The goal of the above described development strategy is to model, design and realise systems using

collaborative engineering. The aim is threefold:

- Reducing the time-to-market
- Less costly prototyping
- Better product quality among others because of end-user involvement

Collaborative engineering means organisational and technological support for multidisciplinary integrated design with many people working at different locations. The Manus manipulator [3], the commercial available general purpose rehabilitation robot, is at this moment re-engineered following these principles in the Commanus project [4]. Elements of the method are applied in the Mobinet European TMR project.

Visualisation turns out to be very important in multidisciplinary designs. Mono-disciplinary views on (simulated) device models answer each moment the question : Are we still working on the same robot? In Mobinet mechanical rapid prototyping (Lund University and BIBA Bremen) control rapid prototyping (TNO-TPD and University of Reading) as well as embedded system integration rapid prototyping (TNO-TPD) is addressed. The latter one is also dealt with in the TIDE ICAN project [5].

User-interface design is extremely important for end-users. The web based ProVar approach [6] and the Manus adapticom method [7] in the Netherlands can be mentioned.

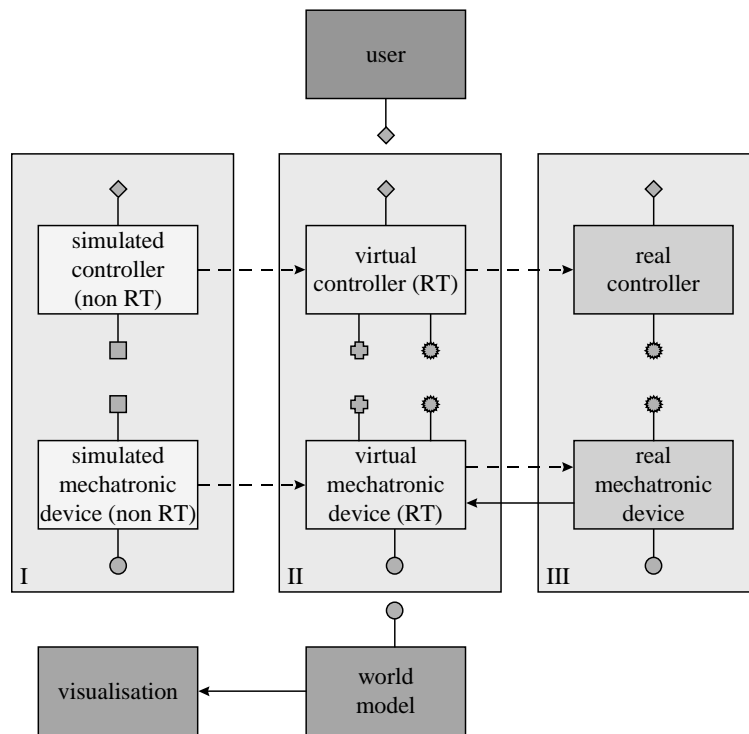


Figure 1. Rapid prototyping for mechatronic systems.

RETIMO: A rapid prototyping environment for assistive devices.

Structure

The structure of the RETIMO development environment is shown in Figure 1.

The big blocks represent the simulation phase (I), the virtual prototyping phase (II), and the prototype realisation phase (III). The small blocks represent system components, such as the controllers, the dynamical model, the world model, etc. The blocks can communicate with each other using interfaces, represented by lines ending in a small shape. Blocks can only communicate if they have identical interfaces.

The user can generate commands to the controller. The control exists either in simulation, or as a virtual prototype, or in reality as a real prototype of the embedded system. The *simulated* controller, can only communicate with the *simulated* mechatronic device (since they or both non-real-time). The *virtual* (real-time) controller can communicate with either, the *virtual* mechatronic device, or with the *prototype* of the mechatronic device. Note that in all stages (simulation, virtual prototype, or real prototype), the mechatronic device has an interface towards a visualisation environment. This means that an end-user can see how the total system will behave at an early stage. Especially at the virtual prototyping phase, he/she can already evaluate the system or practice with it.

Embedded system prototyping

Designing the embedded system means, among others, taking decisions about the real-time environment (VxWorks, PharLap, WindowsCE), the organisation of the real-time processes (number of parallel threads), the bus type, and the distribution of the controllers over the real-time threads. Based on UML, we are able to identify efficiently the specifications for the real-time system. Using Rationals Rose [8], we can create real-time environments for different operating systems, meeting the requirements found during the system analyses. Interfaces between the real-time environment, and the Matlab/Simulink Real Time Workshop [10] exist, so controllers can be efficiently merged in the embedded system (see below).

A virtual communication bus exists between controller and mechatronic device. The virtual bus can be configured as a CAN bus, or a USB bus, or a serial port. Interfaces exist between the virtual bus, and the corresponding “real” buses, making it possible to communicate with virtual prototypes and real prototypes at the same time (hardware in the loop simulations).

Controller prototyping

For developing control algorithms we use the Matlab/Simulink simulation environment. Matlab offers a wide variety of design tools for different types of controllers. Once we have satisfactory simulation results, we want to create a

real-time controller, without re-coding the designed algorithm. For this we use the Matlab/Simulink Real Time Workshop (RTW). A disadvantage of the RTW is that it generates only one single C-function, containing the functionality of the entire Simulink model. When this model contains several control blocks, all blocks are combined into one C-function, which is very inconvenient for developing a hierarchical or distributed controller. This problem can be solved by writing the controllers directly in C, as a so called sfunction. Matlab tools can be used for optimising the controller, and after code generation, the different controllers can easily be identified. Also during controller prototype generation, the C-algorithm can be reused.

The virtual controller can communicate with the real mechatronic prototype, enabling hardware-in-the-loop simulations. Here, some components of the device are virtually prototyped, whereas others exist in reality.

Visualisation

Currently we use OpenInventor for visualising the world model. OpenInventor is a tool which is built on top of OpenGL. 3D Objects can be created using 3D Cad packages, and exported to OpenInventor. At this moment we do not have the possibility to interact with the world model during simulations. This functionality would be useful, since it can help in investigating the response of the system on an unexpected event (e.g. placing an object in front of a mobile base).

Designing a mobile base mounted manipulator

The system.

In RETIMO, we design a mobile manipulator, which will be used by elderly and disabled persons for carrying out all day living tasks. At TNO-TPD the mobile manipulator is composed of the Manus manipulator, and a LabMate mobile base [9]. A picture of the two subsystems is shown in Figure 2.

The controller

For developing the control system of the manipulator, the figure below shows the results of a joint speed controller, which was designed in Matlab. The speed

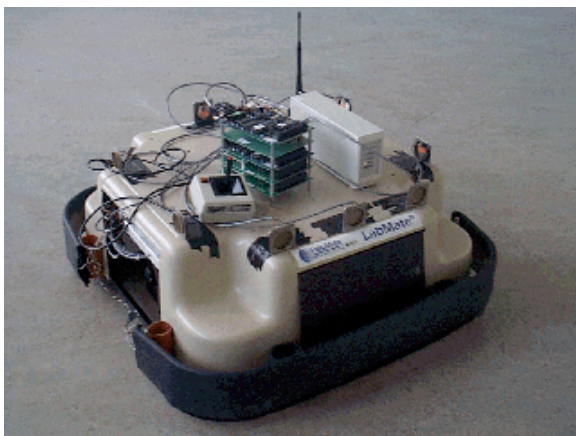


Figure 2. The Manus manipulator, and the Labmate mobile base.

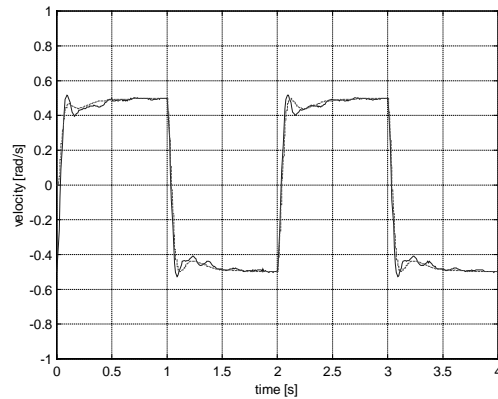


Figure 3. Simulation results and HIL results.

controller was compiled to a virtual prototype. We tested the virtual controller with a real prototype of the Manus (hardware-in-the-loop). In Figure 3 the results of the HIL simulation and the Matlab simulation are shown.

As can be seen, only small differences between simulation and reality exist. For this situation, new developments can indeed be tested on the virtual system, since this behaves with the same dynamics as the real system.

The visualisation

We've build a visualisation of the entire system for showing how the total system will look like in practice. The results of the visualisation are shown in Figure 4.

Future developments

RETIMO has proven to be powerful in speeding up developments. User involvement needs to be more intensive. Provisions are made for interfacing to standard powered wheelchairs. Current activities are the development of a RETIMO-M3S interface. In the TIDE-ICAN project, interfaces to DX as well

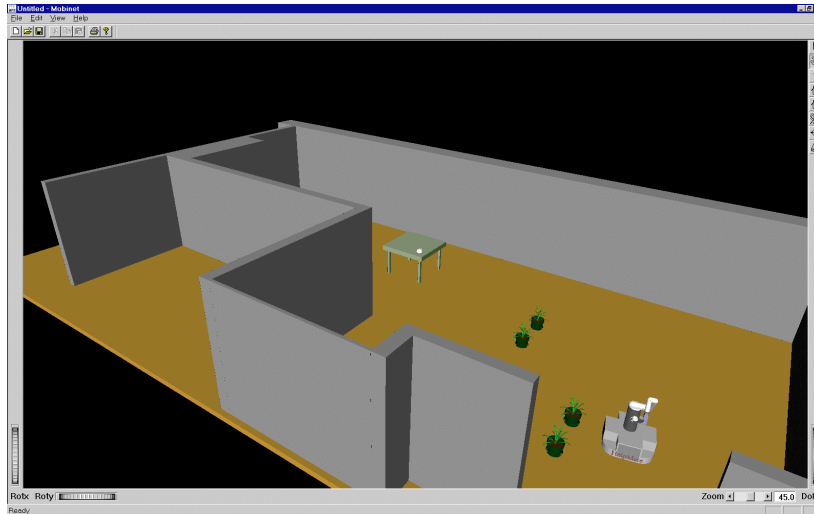


Figure 4. Visualisation of the total system.

as P&G are developed. In this way RETIMO can also interface with these industry standards.

The next step is the ability to present virtual simulation results over the internet and be interactive with users. For the ProVar workstation [6], this functionality is already partly available. We believe that enabling the virtual prototype functionality over the internet will strongly increase the demand and the acceptance of using (advanced) assistive devices by persons where they are meant for.

References

- [1] **MobiNet.** "Mobile Robotics Technology for Healthcare Services". A TMR project. Project nr: FMRX960070. Web: <http://147.102.33.1/mobinet/mobhome.htm>.
- [2] **Burns, Marshall.** "Automated Fabrication - Improving Productivity in Manufacturing", Prentice Hall, Inc., New Jersey USA, 1993.
- [3] Manus manipulator information package. January 29th 1998. Web: <http://www.worldonline.nl/~dynamics>.
- [4] **COMMANUS** EU-CRAFT (BIOMED 2) project nr. BMH4-CT98-9581.
- [5] **ICAN** EU-Telematics 2C. Project nr. DE4204.
- [6] **ProVar** home page: <http://provar.stanford.edu/>
- [7] **Kwee, Hok.** "Integrated control of MANUS manipulator and wheelchair enhanced by environmental docking." *Robotica* (1998) volume 16, pp. 491-498.
- [8] **Bruce Powel Douglass.** "Real Time UML. Developing Efficient Object for Embedded Systems". Addison-Wesley. ISBN 0-201-32579-9. 1998.
- [9] **Labmate** mobile base. Web: <http://www.ntplx.net/~helpmate>
- [10] **Matlab RTW user's guide.** The MathWorks, inc. May 1997. Web: <http://www.mathworks.com>.