RS Embedded Development Platform Case Study

Simon Morrison, Managing Director, NeuroRobotics Ltd.

1. Objectives

The objectives of this case study are to introduce some of the challenges in developing robotic systems in relation to the electronics, and how the new state of the art RS Embedded Development Platform significantly eases such a development. An example application is described.

2. Introduction

NeuroRobotics is a young and upcoming high tech company founded in 2005 to actively pursue robotics research and commercialise several unique robotic designs – with a focus on telerobotic systems, customised manipulators & haptic interfaces for the service industry. NeuroRobotics works in close partnership with various universities and international partners – including Sussex University Centre for Computational Neuroscience and Robotics (CCNR)

Robotics is a rapidly growing and exciting field that spans most areas of science and engineering. A typical robotics research platform that NeuroRobotics would design and manufacture incorporates many electronic, mechanical and software sub systems that tightly interact in real time within a continually changing and uncertain environment. The technical and commercial challenges involved in developing and successfully deploying such a robot in the field are considerable.

The new state of the art Embedded Development Platform (EDP) from RS eases some of these challenges for the robotics engineer. This paper describes a typical robotic system, how NeuroRobotics integrated the RS EDP into a robot, the resulting benefits gained and the ongoing opportunities within the robotics market.

3. Typical Robotic System & Challenges

Figure 1.0 below shows a NeuroRobotics NeuroArm™ integrated into a mobile robot (ActivMedia) - used here at the Intelligent Systems Laboratory at Imperial College, University of London.

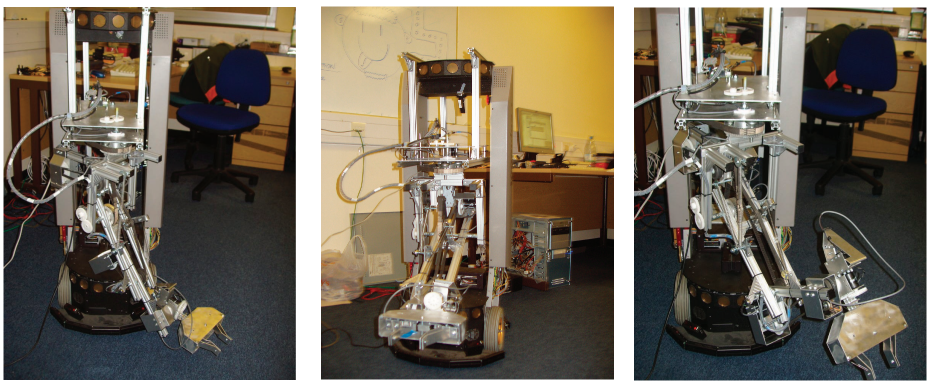

Figure 1.0

The above system is typical of a professional robotic research platform, that would include the following hardware:

<u>Sensors</u>

- Ultrasonic arrays for obstacle detection
- Infra Red proximity sensors
- Tactile bump sensors
- Range finding Laser
- Gyroscopic inertial sensors & Accelerometers
- Tilt sensor
- Compass
- GPS
- Wheel odometry sensors
- Arm joint position, velocity & torque sensors
- Camera system
- Microphone array

<u>Actuation & Motor drives</u>

- Dual drive motor system for mobility
- 6 motor drives for robot arm
- Gripper mechanism

Electronics

- Obstacle detection & avoidance electronics
- Locomotion control electronics
- Navigation electronics
- Robot arm drive electronics
- Speech synthesiser electronics
- Speech recognition electronics
- Vision processing electronics
- Battery charging and power management electronics
- Dynamics and Inverse kinematics processor
- High level Host processor for AI / Robot applications

Human Machine Interfaces (HMI's)

- Joystick
- Haptic
- Speech recognition
- Key pad & LCD display

Communications

- Wifi
- USB
- Bluetooth
- RS232

4. Factors effecting choice of hardware

To fulfil the required robot function - the architecture of the above hardware sub-systems needs to be designed and in turn orchestrated by several levels of software.

There are many ways to implement such an architecture including a hierarchical, flat, distributed and centralised topology or a mix of these. Whatever architecture is chosen, it is clear from the above list of hardware sub systems – that the resulting system will incorporate a large number of electronic circuits that need to be powered and communicate with each other.

raditionally the roboticist would either select the electronic sub system from an appropriate vendor or design their own bespoke unit. The choice of all above electronic subsystems would depend on several factors:

- Available functions
- Performance
- Price
- Delivery time
- Physical dimensions and weight
- Fixing methods
- Ease of electrical connectivity and connector types
- Is it in kit form or assembled
- Need for soldering
- Hardware technical support
- Software support and availability of libraries & tools
- Open source or proprietary software interfaces
- Ease of switching in another supplier
- Future proof capabilities
- Reputation & brand
- Compatible communication interfaces
- Learning curve requirements
- Favourite supplier
- It's the only company that has one
- One is available in the lab to use now

The importance of the above factors and their corresponding priorities will depend on whether the project is being carried out by a commercial enterprise, a university research department or indeed a robotics enthusiast – the latter being a rapidly growing market.

5. Robot Implementation prior to RS EDP & typical challenges

Before the RS EDP availability – the above robot system would comprise of electronic hardware from about 17 different companies and some in house designed circuitry. Each of these circuits would typically use different connectors, different fixing methods with a large amount of time making up cables / connector arrangements. The units based on processors, would in the main, all have I2C or UART interfaces which simplify things – but would invariably run with different operating systems that in turn would use different software development & debug tools.

Figure 2.0 below shows a more general schematic block diagram of one of NeuroRobotics systems that incorporate multiple vendor sub systems.

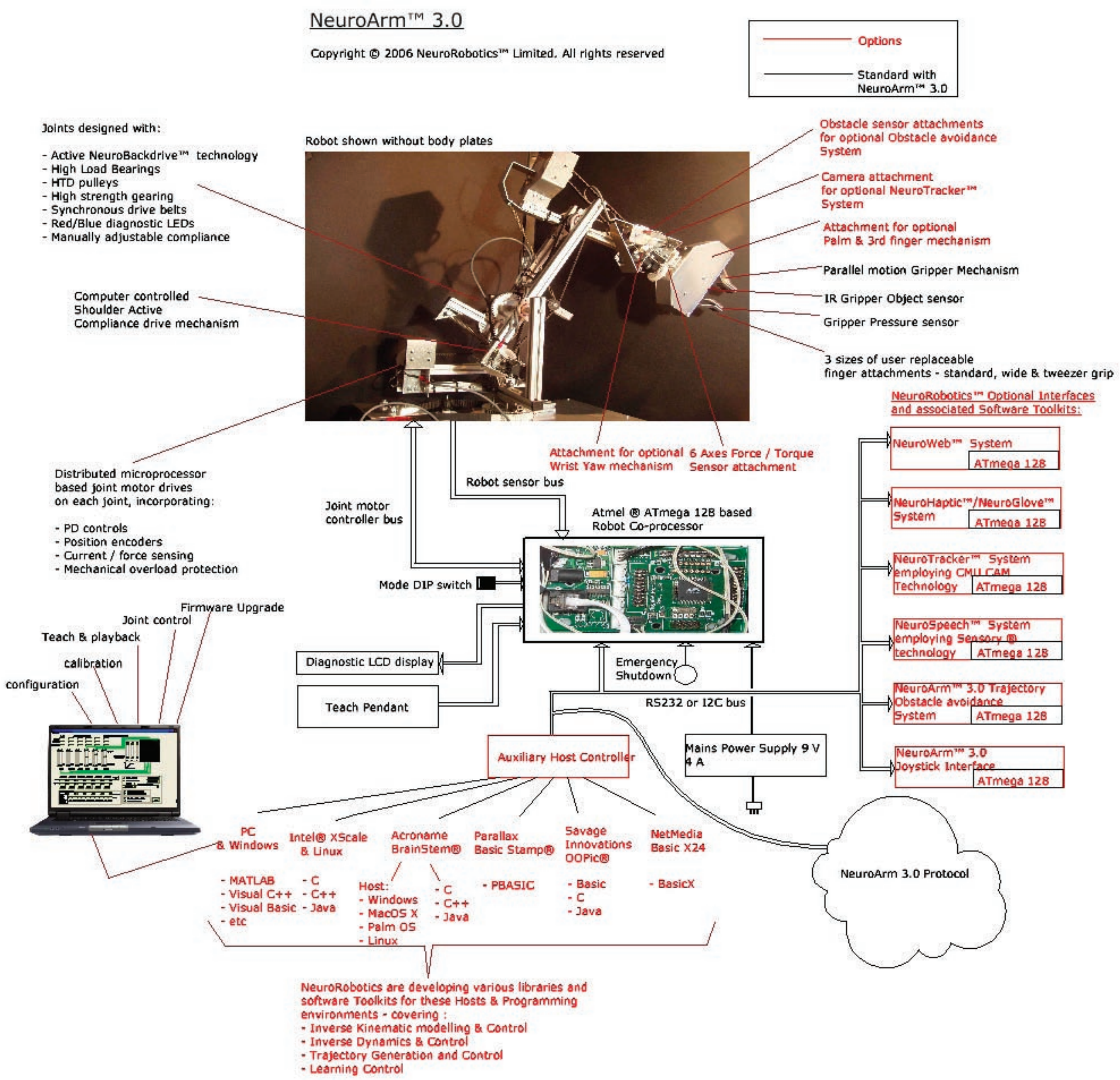

Figure 2.0

Some of the systems would be programmed in C, Basic, Java or vendor variants of these. Some would be configured in different modes with software, some would use jumper switch configurations and some DIP switch settings. Once the system is wired up with all the necessary power and communication cables looms, the development of the software requires multiple software tools running at once – some windows based, some command line based running in a DOS shell. Apart from a myriad of device drivers that need installing, care needs to be taken in which order various tools are installed on the PC, which tools can operate simultaneously or indeed which tools can actually reside on the same PC.

Unless you have a multi-disciplined team dedicated to each sub system that individually have the respective expertise – the high level architect tends to be one or few people that would need to know all these systems.

During development and debug, electronic circuits sometimes need to be removed for debug or jumper switch modifications - which can be problematic if what you need to access has multiple connectors and cable feeds for I2C, SPI, analogue power, digital power, Digital IO, RS232, JTAG debug terminals and an array of motor and sensor wiring. Apart from the EMC challenges of such a setup – continual unplugging and plugging in of connectors can result in static damage, poor connections, placing strain on the PCB and also the danger of plugging the wrong connector into the wrong socket.

Unless the roboticist actually enjoys the above process, and some actually do, most just need to dive into the actual nitty gritty robotics and artificial intelligence development as soon as possible.

6. Robot Implementation with the RS EDP and key benefits

Figure 3.0 below shows a Mobile Robot platform integrated with an RS EDP, and Figure 4.0 below shows a block diagram schematic of a current robotic platform being developed for research being carried out between NeuroRobotics and Sussex University. A similar robotic platform is also being designed and manufactured for RS as a demonstrator and a more in depth case study will be published on this system when it is complete.

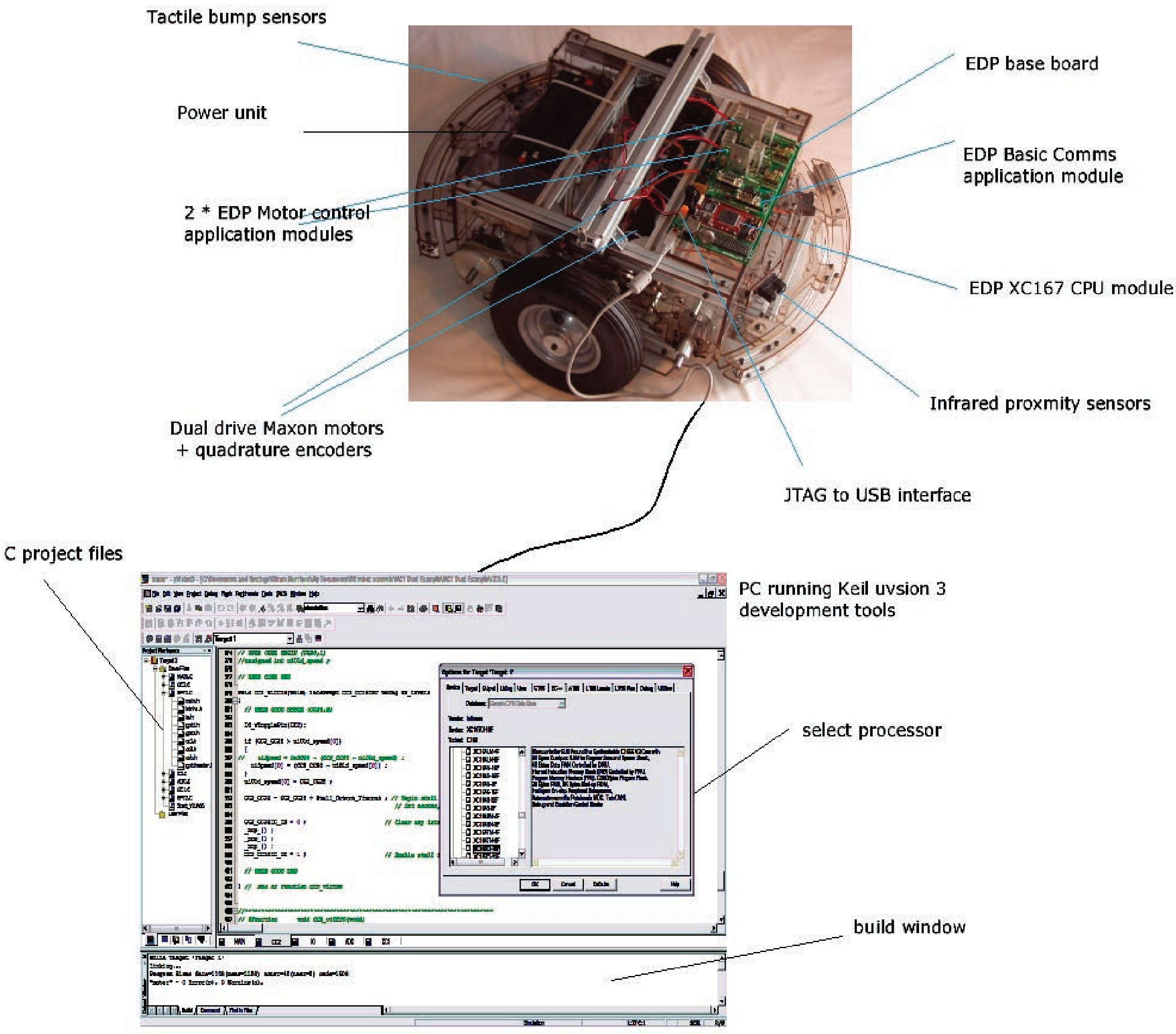

Figure 3.0

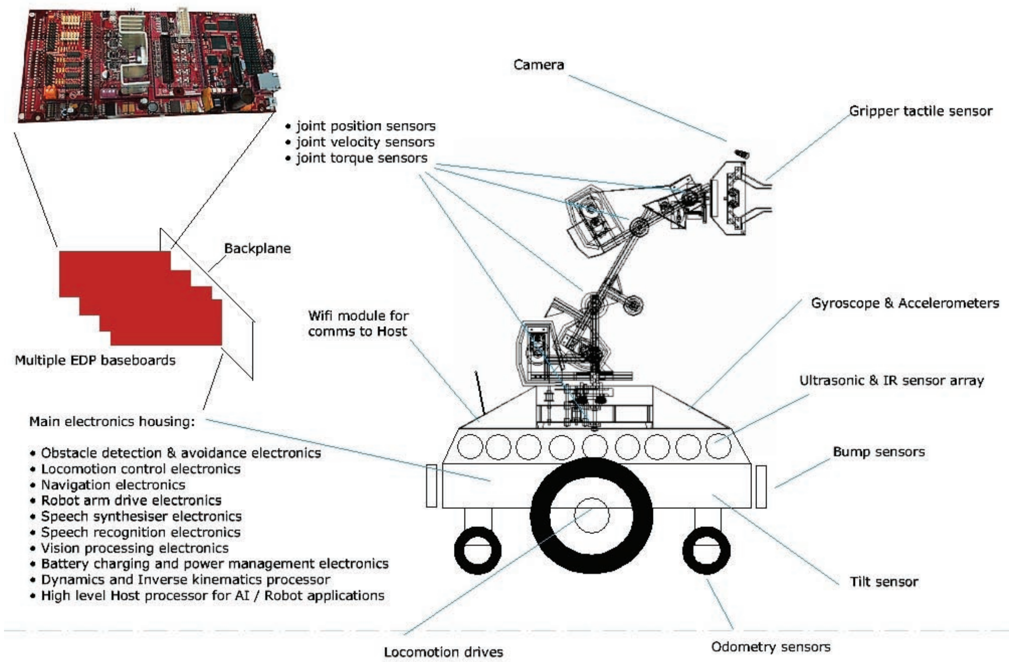

Figure 4.0

Robotics development is already multi-disciplined in that the engineer/team needs to be competent in:

- Mechanical engineering
- Software engineering
- Artificial Intelligence
- Electronics & Electrical Engineering
- Mathematics
- Dynamics & Statics

The RS EDP goes a long way to considerably removing a lot of the aforementioned challenges thus enabling the roboticist to more quickly and efficiently get on with the robotics application.

The first application that NeuroRobotics developed with the RS EDP is shown in Figure 3.0 above. The robot required 2 motors drives + quadrature encoders, and obstacle sensor processing over an I2C bus, a 16 bit processor that was fast enough to run various motor control and navigation algorithms and suitable software tools that enable coding, debug via JTAG and flash download facilities.

We chose the following components of the RS EDP:

- EDP Base Board
- EDP Motor Control application module
- EDP XC167 CPU module based on the Infineon 16 bit Microcontroller
- EDP Basic comms application module
- Keil uVision 3 software tools

After next delivery from RS – we were able to install the software, plug in the modules, wire up to the robot motors and sensors and start testing some basic code fairly quickly. As this was the 1st time we used this system – some learning curve needed climbing including jumper configurations and software installation sequence of the device drivers required (JTAG to USB interface) and the compiler. Any technical questions on the hardware and software, along with support on the C software libraries were dealt with swiftly by RS/Hitex engineers – and the basic software and simple motor tests were up and running surprisingly quick. We had a few teething problems with flash download and using the JTAG debug facilities – that were solved by following carefully the instructions of boot up sequence and timings. Apart from that the only time consuming factor was making up cables & connectors for the motor and quadrature encoder to plug into the EDP Motor Control application module. This is now overcome by having sets of standard cables.

The Keil software development tools are extremely professional, and by setting up a project that contains all the processor parameters, flash download configuration and C library setup – each new development session just automatically sets up the respective parameters and loads all the necessary C files ready to jump straight in were you left off.

The quality of the connectors are automotive standard and lend themselves to robots that suffer vibrations from motors and driving over rough terrain – as well as lending themselves to applications intended for the outdoors.

No time was wasted with any time consuming inter module wiring. The EDP baseboard enabled us to quickly switch over to using the other processor board EDP STR912 CPU module based on the ST Microelectronics STR912 Microcontroller – by virtue of the novel mapping technique of the EDP base board. The EDP mapping technique forms the core of the EDP Virtual CPU concept that allows several different processor modules to appear functionally transparent in terms of the various buses (Address I2C, SPI, CAN), IO (digital & anlogue) & power lines.

At the beginning of this paper – the multitude of processor based electronics that reside in research based robotic systems will benefit enormously from this virtual CPU concept to essentially enable the same specialised processor core to be utilised – but using an application module that is designed to meet the EDP specification. In addition RS provide the gerber files and various other data on an IP free basis – resulting in a rich variety of application modules being currently developed with a route map of modules being introduced on a monthly basis. This also means that if a module isn't available – the user could customise their own.

To implement the robot as shown in Figure 4.0 – several EDP base boards will be required to house the multiple application modules. This will be realised by attaching 64 pin DIN414162 connectors to each base board and then in turn plugging each of these into a backplane.

7. Robot application modules for the future

Modules that are of particular interest to us within the robotics field, some of which are on the RS route map, are:

- Multi channel (minimum 6) DC brushed motor drives
- Multi channel servo drives to control traditional radio control style servos
- SD Card & compact FLASH module
- Inertial sensor unit for robot navigation - compass, gyro & accelerometers
- Vision processing module
- Force / torque sensor processing module
- Intel Xscale 32 bit processor module
- GSM module + SIM card slot
- Keypad & display module
- Dedicated obstacle detection and proximity sensing module
- Haptic joystick processing module
- Bluetooth
- Zigbee
- Wifi
- Speech recognition & Speech synthesiser
- Battery management module with Photovoltaic input and charger

Some traditional robotics based hardware based on the intel XScale, OOPic, Basic Stamp & Brainstem to name a few - may actually be better suited to sit at the core of some of the above robot application modules. However, there will be a trade off between having to use some of the pre existing development tools vs. transparency they offer up to the EDP baseboard by capitalising on the Virtual processor concept. There are several ways around this including the integration of the Microsoft Robotics Studio and developing application modules with software that are compatible – however, this is beyond the remit of this paper.

8. NeuroRobotics ongoing developments with the EDP

Some of NeuroRobotics robot developments to date are based on the Korebot embedded PC hardware running Linux supplied by the well established K-Team, the widely used Webots Simulation & Modelling tools from Cyberbotics and the popular Real time Programming tool URBI from Gostai. We are also developing a MATLAB robotics toolbox that will enable mathematical modelling, analysis and the generation of some very complex robotic control algorithms – in particular on the non linear time varying dynamics of the robot arm.

After seeing the benefits of the RS EDP and RS's commitment to a route map long into the future for this system, we have commenced migration of these powerful software tools to provide compatibility with the RS EDP offering – thus giving our customer base the choice if they wish. Figure 5.0 below illustrates such a system

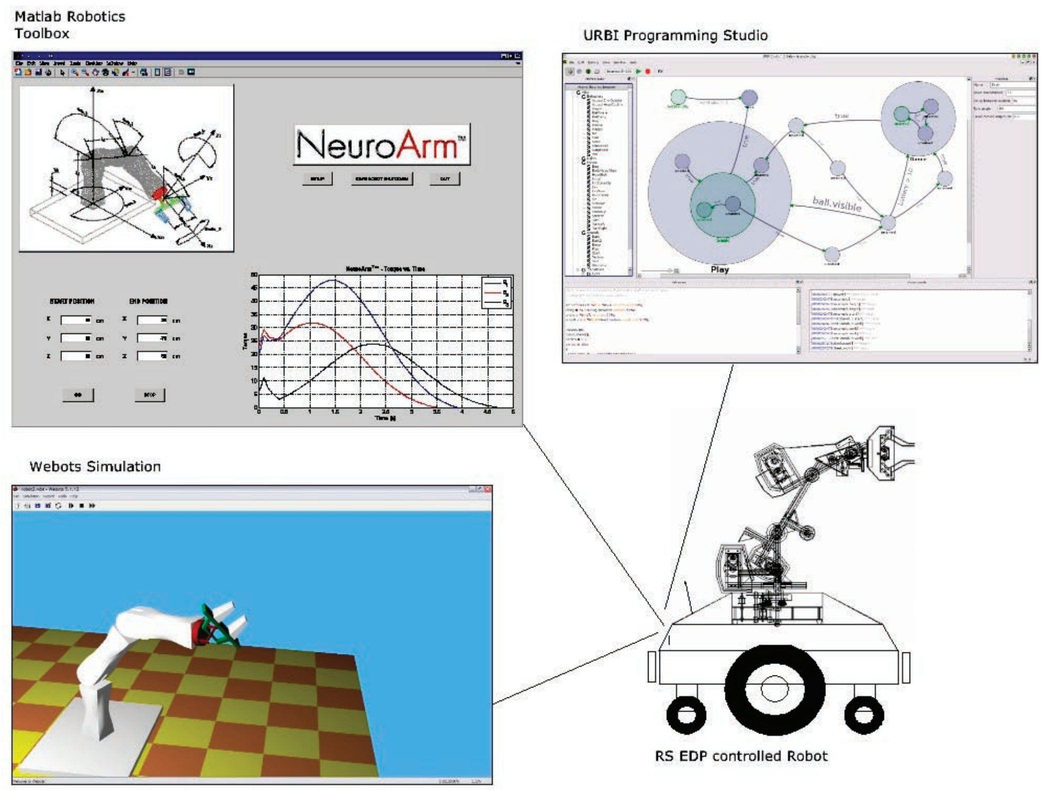

Figure 5.0

A separate paper is being published that explains in some detail what these tools will be able to do, and how they will be interfaced to the RS EDP.

Conclusions

With the market flooded with literally hundreds of embedded hardware and supporting software tools, the prospect of yet another system for the robotics engineer to evaluate, learn and gear up for – didn't seem that appealing. From a business perspective, the cost of the engineers learning a new system, purchasing new kit, migrating existing hardware & software, and the associated risk with an unknown/ untested solution - also is clearly a barrier to change.

However, after reviewing the new RS EDP, it became clear very quickly that this was not just another 'me too' system – this was a new concept. From a purely commercial perspective - this concept can save the company money, time and deliver substantial value added functionality & performance to help boost profits. From a purist robotics perspective – this system opens doors to more rapidly develop the robotic system. This is because upon closer examination a large number of the factors effecting the choice of hardware described in section 4 – are weighted in favour of the RS EDP.

If RS continue to develop this system and furnish the engineer with an extended range of application modules, offer further software tools and continue on the path to developing a more structured set of documents and technical support base – this system could become the choice for roboticists on a large scale.

To find out more about the NeuroRobotics robots described in this paper, and ordering an RS EDP robotic system - please visit http://www.neurorobotics.co.uk

NeuroRobotics Ltd • Sussex Innovation Centre • Science Park Square
Falmer • Brighton • BN1 9SB

Head office: +44(0)1273 704784 • Email: sales@neurorobotics.co.uk