The Development of a Low Cost approach to the Prototyping of Construction Plant

D A Bradley[†], F Margrave[†], M B Widden[‡], G Melling[#] & H. McKee[‡]

[†] School of Electronic Engineering & Computer Systems, University of Wales, Bangor, Gwynedd LL57 1UT, UK
[‡] Engineering Department, Lancaster University, Lancaster LA1 4YR, UK
[#] Consultant

email
D.Bradley@sees.bangor.ac.uk
M.Widden@lancaster.ac.uk

Abstract

Because of the high development costs of an item of automated or robotic construction plant there is a need for a low cost rapid prototyping system which can support the early stages of the design and development process and provide feedback to the designer of the effect of different kinematic architectures on performance. To be fully effective this same system should also provide for the support for other areas of the design process including support for the design and evaluation of the user interface and for software design and evaluation, including code generation.

The paper describes the development of such a rapid prototyping system and illustrates how it can be used in support not only of the general design process but of other areas such as task evaluation.

1 Introduction

In the design of large items of plant and equipment such as those used in the construction industry, sub-sea applications and by the military there is an increasing need to provide the designer with the means of rapidly evaluating the operation of the plant within its intended operating environment and of comparing alternative design configurations [1,2]. However, for systems on the scale of a typical item of construction plant, the cost of developing a prototype can be high and the cost in a competitive market of introducing significant structural and kinematic changes into that prototype prohibitive.

Further, while sophisticated computer simulations have been developed and used by industries such as aerospace, particularly for military systems, the cost of the computer hardware and software for such systems is still extremely high and beyond the reach of the majority of commercial systems designers. Their use also tends to require that the basic system is well defined which may not always be the case [3].

For these reasons, among of course many others, the designers of construction plant may well tend to be over conservative and hence fail to take effective advantage of the opportunities offered by the application and use of computer technology in the design process [4,5,6].

There is therefore a need for the provision of a low cost approach to the design and prototyping of large and complex plant such as that found in construction which can be used by the design team from the very earliest stages of the design and development process. Such a low cost system would not only provide the design team with a means of visualising the operation of the plant in a range of possible environments but should provide support in areas such as software development and user interface design. Any such system should therefore be flexible, simple to modify by inexpert users, easy to use and understand and be based on readily available technology.

The paper describes the development and testing of a PC based environment which meets the above requirements and considers how the resulting system can be integrated within the overall design process.

2 System Requirements

The design process of any integrated and mechatronic system such as a construction robot requires the integration and validation of a range of technologies. However, conventional methods such as the building of full scale prototypes are often ruled out by their cost. In order to provide effective support for the design process any system must support the reproduction within the model of the major functional requirements of the full scale system. These include:

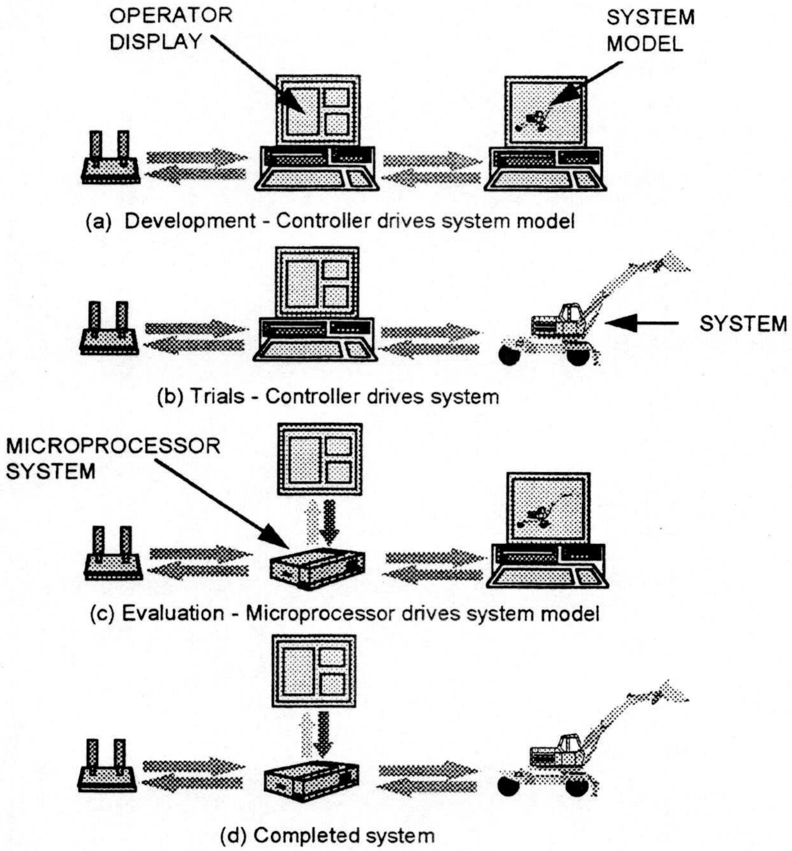

Figure 1: *The prototyping process*

- The ability to rapidly reconfigure the model, for instance to take account of changes in system geometry and kinematic structure.

- The ability to provide the full range of controlled motions taking into account any operator defined limits on the operating envelope.

- Support for the definition by the user of the allowable operating envelope for and the setting of limits on operation for the machine and the orientation of its end effector within those user limits.

- The investigation and evaluation of different operator strategies including a wide range of input mechanisms such as multiple joysticks, multi-degree-of-freedom joysticks and others as appropriate.

- The simulation of a wide range of operator displays and interfaces structures.

- The provision of feedback to the user on the machine and current task status.

- Support for the development of the required operating software.

- The ability to provide the designer with a means of visualising of the operation of the system from a range of viewpoints, including that of the operator.

On the basis of this evaluation of system requirements, it is suggested that the proposed rapid prototyping facility is based around the following major sub-systems:

- A controlling sub-system dealing with the operator interface, displays, machine kinematic models and control algorithms.

- A controlled sub-system capable of emulating the plant and its operating environment at an appropriate level of resolution and providing performance feedback including visualisation.

A typical design cycle would then would then proceed as suggested by figure 1, beginning with an evaluation of the overall system, including the task environment, to assess factors such as the effects of different control strategies and machine geometries on operation and the configuration of the operator interfaces. During this development phase the controlling sub-system acts to drive the system model which is running on the controlled sub-system while the operator display is simulated on the controlling sub-system. Operator inputs are provided to the controlling sub-system by whatever means is required, for instance joysticks.

This arrangement permits the rapid modification of the system, including the provision of links to CAD packages used by the design team for the generation of the plant model. The plant kinematics are held in the controlling system and used to generate the signals to operate the system actuators which are then passed to the controlled sub-system in the appropriate format.

Thus in the case of a hydraulically operated system the information transferred would take the form of the commands required to operate the hydraulic valves to be used on the full scale system. The controlled sub-system then interprets these commands and reproduces the intended motion in the model environment. Information is then returned to the controlling sub-system in the form of

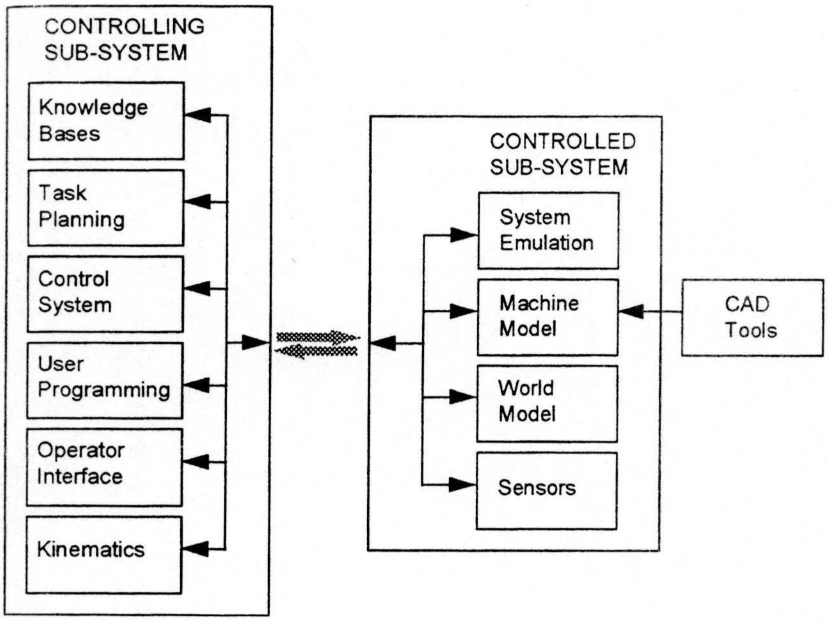

Figure 2: *Distribution of software functions between the controlling and controlled sub-systems*

the equivalent sensor output representing, for instance, joint angles.

At this stage, visualisation of operation should be provided and the implications for software and associated hardware assessed for various task environments. To support the ultimate replacement of the emulation with the real machine, the message passing between the controlling and controlled sub-systems should be of the same form and format as that required by a real machine and the system should therefore be capable of being structured to accommodate a range of message formats and protocols such as CANbus. Figure 2 suggests the distribution of software functions between the controlling and controlled sub-systems.

Consideration can also be given to the design of the operator interface an this stage with the simulated interface being reproduced on the controlling sub-system display. Figure 3 shows a typical operator interface evaluation screen generated during the prototyping process.

Once the design and operational procedures have been developed, these can be confirmed and refined in the next phase of the design process by using the controlling system to drive the prototype plant or a scale model directly and the performance of the software tested against the performance achieved. Further evaluation of the operator interface and displays can also be made at this stage.

Once the performance levels have been established and confirmed , the software can then be transferred to the target processor which can then be used to drive the simulated system as in figure 1c to establish its performance before implementation on the real machine when a full system test can be carried out.

As with any design process, it must be remembered that this simple sequential progression is unlikely to occur in practice and there will inevitably be a degree of iteration and parallel development, for instance to produce a hardware prototype while still developing software.

3 Implementation

The prototyping system that eventually evolved took the form of a pair of interconnected PCs representing the controlling and controlled sub-systems respectively. In developing this arrangement, care was taken to ensure the proper synchronisation of data between the two systems as any failure in this area would have significant and adverse effects on the operation of both sets of programs. This in turn required careful consideration of the mechanisms of data management and data transfer to be employed. In this context the roles of the controlling and controlled sub-systems can now be considered.

3.1 The controlling sub-system

The role of the controlling system is to provide a rapid prototyping and software development facility in areas such as the evaluation of control algorithms and operational strategies as well as to support operator interface design and development. It is therefore structured around a number of hardware and software sub-systems as follows:

Hardware.

Fast, available, non-exotic hardware with a wide range of software support was considered an essential prerequisite in order to achieve a low cost system with maximum flexibility. A PC with appropriate interface cards was therefore selected as the basis for the controlling sub-systems.

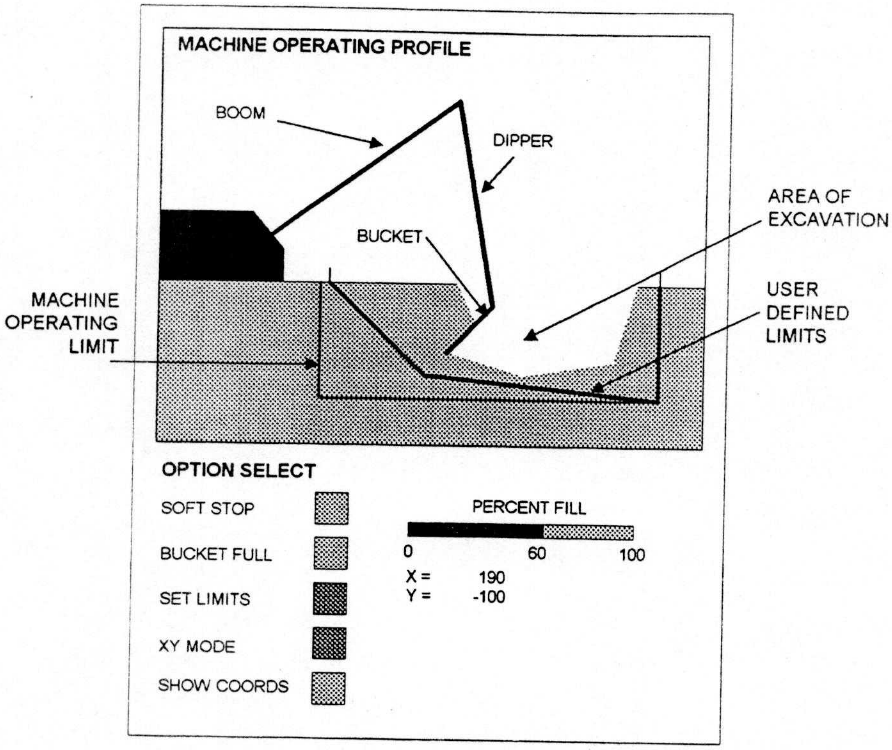

Figure 3: *Operator interface screen*

Software.

In order to support the simulation of the operator's control panel and associated displays, a WINDOWS based system was considered essential and Visual Basic was therefore used to provide a powerful highly intuitive, simple and user-friendly front-end environment for software development. A modular structure was established from the very beginning and the front-end supported the use of other software such as C++ which was used for the underlying modules.

3.2 The controlled sub-system

The challenge with the controlled sub-system was to provide an environment which complemented the controlling sub-system while supporting the modelling of the machine kinematics in an easily assimilated form. An animated, real-time visualisation was considered to be essential feature of the controlled sub-system as this would provide the means by which effective visual feedback on the performance of the simulated machine was provided. Initial considerations therefore suggested a virtual reality (VR) approach based on the adoption of a '*desk top reality*' environment which could be used to create the *world* within which the model of the system under development would exist.

Hardware.

As was the case with the controlling sub-system, the requirement was for fast, available, non-exotic hardware with a wide range of software support, which again resulting in the choice of a PC together an appropriate I/O board as platform.

Software.

The achievement of update rates of 6 to 12 frames per second requires a fast, dedicated package incorporating display and visualisation features and with direct access to the PC resources. REND386 is a software library that supports and facilitates the construction of virtual reality applications on a PC on a '*free and non commercial use*' basis and which has been modified and released as AVRIL*. This was therefore used in the early stages of the development of the controlled sub-system.

A basic C application skeleton is also provided with the AVRIL library that allows the user to load '*worlds*' and '*figures*'. The user can then navigate through and explore the world, interacting with figures and objects within that world in the process. The controlled system was constructed to be entirely generic in that both the world and the figure are described in independent text files. Alternative approaches now being considered include the use of more advance desk-top reality systems and coupling these with available kinematic modellers to achieve the desired level of performance.

In order to model a real environment and system it was advantageous to link the creation of the virtual world and the machine model with material from other sources such as CAD packages. This linkage will then enables the transfer of appropriate three-dimensional (3D) data either directly from the CAD package or indirectly using an appropriate transfer format. Figure 4 shows the relationships between the various elements required to support the creation of the simulation environment.

---

* A Virtual Reality Interface Library

4 Further developments

The basic prototyping system has been tried out using an automated and robotic excavator as the target application and has demonstrated its ability to support developments in both hardware and software.

In the next stage of the development of the prototyping system it is hoped to provide additional support for the optimisation of factors such as machine kinematics in relation to the defined task environment, support for controller design, evaluation of operator strategies including tele-operation, the design of the operator interface and automatic code generation. In addition, it is hoped to be able to increase the level of interaction between the machine model and its virtual environment, for instance to enable the excavator to 'dig' a 'virtual trench'.

Each of these developments will require enhancements to the current system but it is envisaged that they can be incorporated within the low cost strategy used to date.

5 Conclusions

The development of advanced and robotic plant for the construction and other industries presents significant challenges in areas such as kinematic design, mechanical

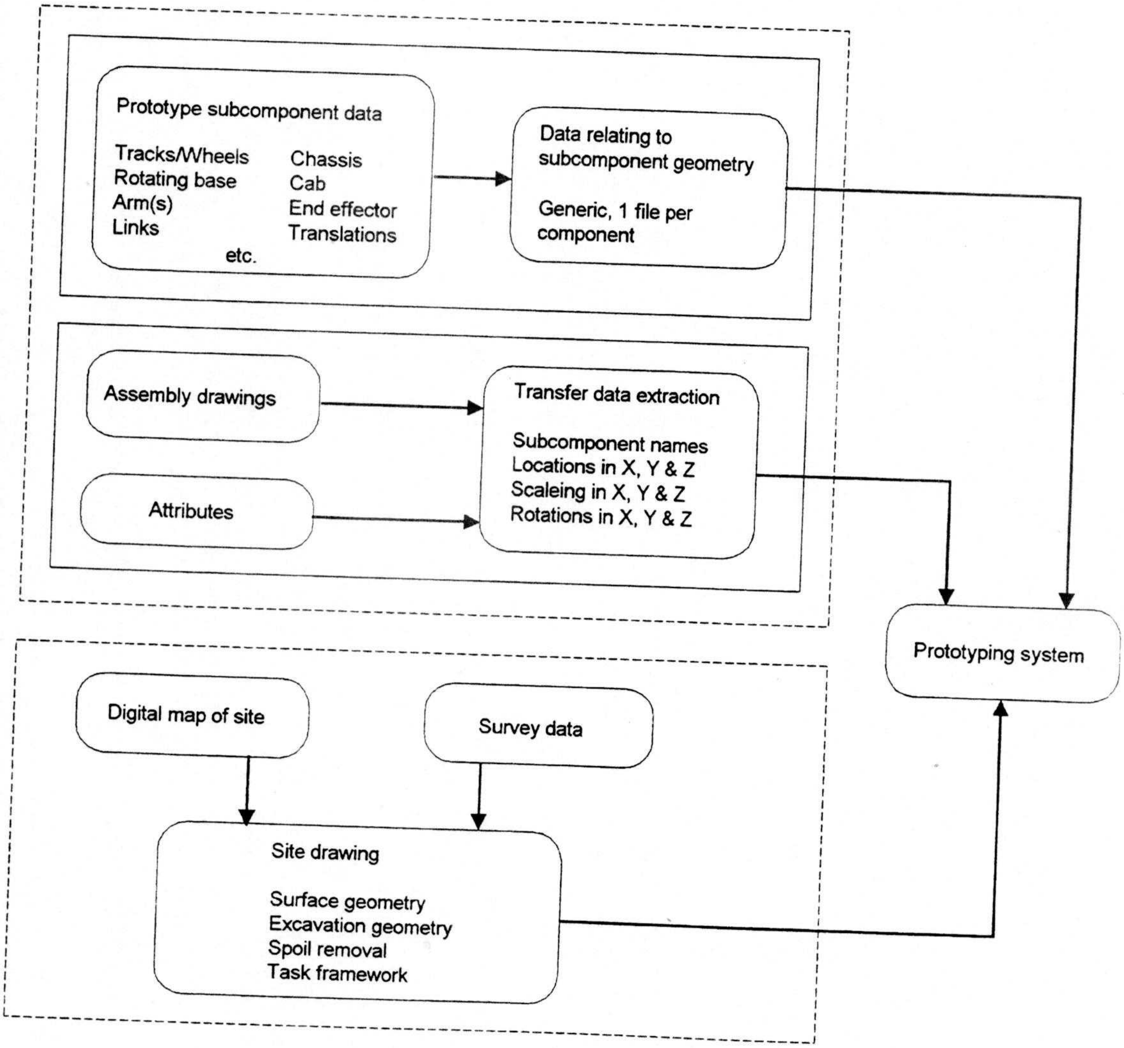

Figure 4: *The simulation environment*

systems and software as well as interface design and operability. Trials have shown that it is possible to build a relatively low cost PC based prototyping system to support a design team in the early stages of the design and development process for large items of plant. In particular, the work to date has shown that the system supports the refinement of the design by including a kinematic modeller and a representation of the virtual world.

Developments of the basic structure to incorporate improved modelling of the environment will enhance the capability while still providing a what should remain a cost effective approach in areas such as visualisation, software design and kinematic design as well as providing the basis for more advanced simulators, for instance for operator training, that may be required.

References

1 Webster, D.A. & Challinor, S.F., 1993, *Long-reach manipulators for decommissioning*, Nuclear Energy, Vol 32, No 3, pp 173 - 176

2 Taylor, A.J., 1993, *Design issues for underwater manipulator systems*, Mechatronics, Vol 3, No 4, pp 419 - 432

3 Staiger, W.A., 1996, *The Growth of Simulation: The M1A2 Abrams main battle tank experience*, Battlefield Systems International, Chertsey, June

4 Thayer, S., Gourley, C., Butler, P., Costello, H., Trivedi, M., Chu Xin Chen & Marapane, S., 1992, *Three-dimensional sensing, graphics and interactive control in a human-machine system for decontamination and decommissioning applications*, The International Society for Optical Engineering, SPIE, Vol 1828, pp 74 - 85

5 Bejczy, A.K., Kim, W.S. & Schenker, P.S., 1994, *The role of computer graphics in advanced tele-operation*, ASCE Conference on Robots for Challenging Environments, Albuquerque, pp 1 - 9

6 Hoffman, R. & Simmons, R., 1994, *Simulation of autonomous robot excavation*, ASCE Conference on Robots for Challenging Environments, Albuquerque, pp 115 - 122