Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

A Mechatronic System Architecture for the development of Flexible Materials Handling Systems

Glen Bright*. Anthony Walker*

**University of KwaZulu-Natal, Durban, Republic of South Africa.*
RSA (Tel: 027-260-1214; e-mail: brightg@ ukzn.ac.za, awalker@ukzn.ac.za).

Abstract: Mass production of customized products requires materials handling architectures to be flexible enough to accommodate varying geometries, masses and volumes of materials. Multi-agent control and coordination structures are under development to provide flexible and scalable transportation systems. In this paper the concept of Mechatronics has been used to develop a Flexible Materials Handling architecture to aid the development of mass customization during product manufacture. Materials handling was provided by the cooperation of multiple differential drive platforms fitted with servo operated and articulated conveyors.

1. INTRODUCTION

Mass production of customised products requires materials handling systems, which are flexible and responsive enough to accommodate real-time changes in materials handling tasks. Architectures developed to implement these materials handling systems require structured development. The architecture must account for all levels of operation, from physical task instances to management systems.

Current AGV systems are designed to operate in singular context, which limits the range of materials transportable to the set contained in the capabilities of a single vehicles materials handling infrastructure. A move towards the utilisation of multiple cooperating AGV systems could greatly expand system scope. Control system and data fusion architectures are under development to aid in producing next generation AGV systems (Bostelman *et al* 2006)(MMHS – Metamorphic Materials Handling System – Final Report, 2003).

Research has been aimed at developing Flexible Materials Handling (FMH) architectures.

The objectives of the research were to:

- Develop a viable mobile platform architecture, which can operate effectively in singular context or during multi-platform cooperative tasks.
- Integrate software and communication mechanisms on to of the mobile platforms, which would allow for generic message passing primitives over a network system.

1.1 A Preliminary Insight

Omni-directional wheels provide mobile robot bases with holonomic kinematics. At low speeds, dynamic influence of the mobile platform can be neglected, allowing the platforms to track arbitrary trajectories. Similar kinematics have been realised via the use of articulated manipulators fitted to nonholonomic differential drive mobile bases (Bhatt *et al*, 2004a, b). When two such equipped mobile bases are attached to a common payload, Fig 1, they have the capability to project lesser-constrained kinematics onto the payload. This allows for more accommodating motions around their working environments. The composite mobile robot configuration has closed loop kinematic constraints however, which must be accounted for when developing motion controllers.

Fig. 1 represents the functional layout of the mechanical architecture used in this research. Relevant degrees of freedom are presented.

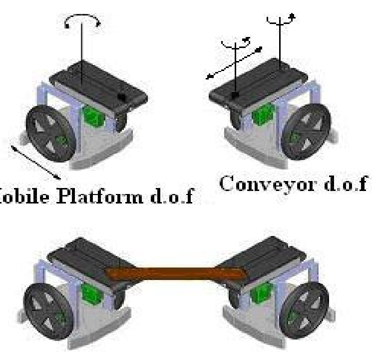

Fig. 1: Diagram of functional layout of nonholonomic mobile bases under cooperative task instance

Each individual mobile platform consists of a differential drive system stabilised by two ball transfer units. Utilising ball transfer units lessens the constraints on manoeuvrability imposed by regular caster wheels (Abou-Samah, M. and Krovi, V., 2001).

2. THE MECHATRONICS APPROACH

2.1 Mechatronics and System Integration

Mechatronics as a discipline is aimed at understanding how system performance changes with subsystem configuration. In this sense it is defined on a differential basis and is used in developing optimally integrated systems.

2.2 The Materials Handling Environment

The manufacturing systems of the future may have vastly different layout configurations compared to current configurations. The layouts of future manufacturing systems will most likely resemble batch and Group Technology layouts. These new layouts would allow for the increase in autonomous elements in the manufacturing environment.

Design of FMH architectures requires an understanding of the kinds of materials handling tasks, which would be required of the materials handling agents. To better the design choices made during the development of the FMH architecture the materials handling tasks were split up into various contexts with respect to the mobile materials handling platforms and the volumes of transported loads. They are:

- Single platform low volume.
- Multiple cooperating platforms low volume with common load.
- Multiple cooperating platforms high volume with common load.
- Single platform high volume.

Current AGV systems carry out the first context in a repetitive manner. It is in the multiple platform contexts that platform architecture requires special attention. Multiple cooperating platforms transporting a common payload should not transfer random forces between one another due to unanticipated and uncorrelated motion errors. The platforms developed during the research project used the articulation of the conveyor to absorb any disturbance forces brought on by motion errors of cooperating platforms. Task instances may require payload transfer between two or more platforms. The conveyor alignment system developed for the mobile platforms projects off exteroceptive sensor information to obtain global alignment. This procedure is also used to update platform orientation, which is discussed in section 6.

3 MOBILE PLATFORM DESIGN

3.1 Mobile Platform Architecture

By considering the various task instances, the platform was designed with the intention of creating a mobile base, which could provide the means for effective operation in both the singular context and during cooperative tasks. The mobile platforms are differentially driven and are stabilised by ball transfer units, providing static balance. The platforms have a servo operated and articulated conveyor system and have the required sensory infrastructures for local obstacle avoidance.

3.1.1 Mobile Platform Design

3.1.1.1 The Drive Subsystem.

All control systems on the platforms are distributed. A dedicated micro-controller runs embedded code to control the speed and direction of the motors thus providing two control inputs, the translation and rotation of the differential drive base. Fig. 2 shows the main components of the mechanical drive set up. One drive motor is removed for full view of support bearing.

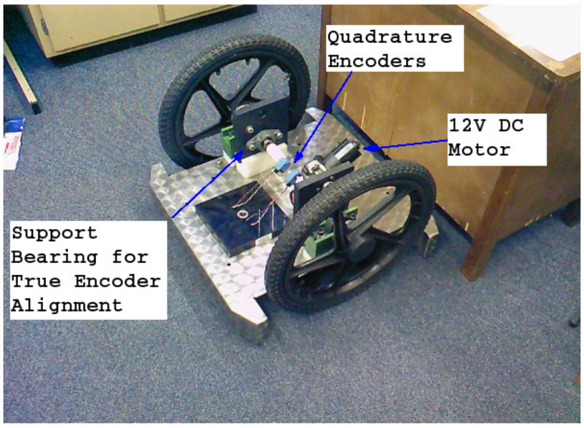

Fig. 2: Mechanical drive system

The electronic drive control module consists of a PIC18C252 based embedded micro-controller namely the BrainStem® Moto 1.0 module (Acroname Inc.) which provides Pulse Width Modulation (PWM) and direction signals to two 20 Amp H-bridges. 256 counts per revolution quadrature encoders provide a feedback control signal. The BrainStem® Moto 1.0 module has efficient PID control loops embedded in the module which can be configured between position and velocity control dynamically whilst the platforms are operational. Although one can obtain motor control modules with integrated motor drivers, separate H-bridges were used as they interface directly with the electrical characteristics of the geared DC motors. This allows system upgrades without breaking software infrastructures. This produces less downtime in industrial settings.

3.1.1.2 The Servo Conveyor Subsystem

The operational scope of the conveyor subsystem was split into two contexts. In a singular context the conveyor retrieves and dispatches small and medium sized materials between machine tools and other mobile platforms. For those purposes it acts as a multi-directional material conveyor. In the context of multi-platform cooperation during the transport of larger materials the conveyor subsystem is automatically configured to articulate thus providing arbitrary end connector trajectories, to be discussed under section 3.2. The conveyor belt and conveyor frame are driven and rotated by two 12V DC motors which are controlled by another PIC18C252 based embedded micro-controller namely the BrainStem® GP 1.0 module. The motors are driven via an IIC based motor driver which is housed on the IIC bus of the BrainStem® GP 1.0 module. The conveyor slides forward on linear bearings allowing for multi-platform material transportation. The conveyor rests on a rotary housing and thrust bearing assembly. Both the thrust bearing assembly and the end connector assembly are fitted with quadrature encoders to provide conveyor and end connector orientations. This provides sensed articulations (Abou-Samah *et al.* 2006).

3.1.1.3 Computing and Data Acquisition Infrastructure

The platform uses as its main onboard computer, a Mini-ITX form factor motherboard with the VIA C7 CPU. The main onboard computer manages the various embedded controllers via 'status and request' messaging rather than continuous raw data communication. A data acquisition board monitors the various battery voltages on the mobile platforms and is used to signal low power thus starting a recharging routine. The onboard computer houses higher level software, which is discussed under section 4.

3.1.1.4 Sensory Infrastructures

An array of 12 ultrasonic transducers, provide the platform with exteroceptive sensor data, this data is used for local obstacle avoidance. The obstacle avoidance software mechanism will be discussed in section 4. The ultrasonic sensors are placed 30º apart and form a 360º ring. The firing sequence of the ultrasonic sensors is such as to try and minimise crosstalk between the sensors.

Other sensors making up the platforms sensory infrastructure include an Inertial Measurement Unit (IMU) and a digital compass fitted to the underside of the conveyor. The significance of the digital compass is discussed under section 6.

3.1.1.5 Communication system

The platforms communicate with one another and other host computers over a wireless network. Each platform is equipped with an IEEE 802.11g wireless network adapter. Communication is performed through TCP sockets.

3.2 The Kinematic Environment of the Mobile Platforms

A description of the kinematic environment of the mobile platforms is included here in order to provide insight into the choice of platform architecture.

Many control architectures have been developed to produce holonomic end connector trajectories in the constrained domain of nonholonomic mobile bases. The control architecture used here is derived from (Abou-Samah *et al.* 2006). Fig. 3 depicts one mobile platform with the conveyor bearing mounted at the mid-point of the wheel axle.

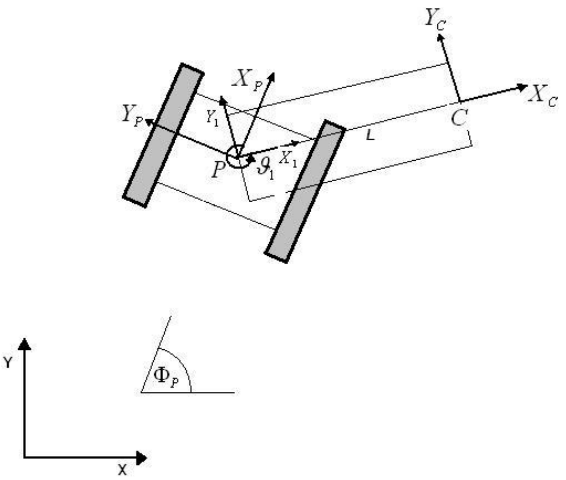

Fig. 3: Diagram of mobile platform and servo conveyor.

Reference frame {P} is rigidly attached to the mobile platform with the *X*-Axis in the direction of forward translation. Reference frame {1} is rigidly attached to the first revolute joint, i.e., the thrust bearing rotary housing. Reference frame {C} is rigidly attached to the connector, which is just a second smaller thrust bearing and bar assembly. The configuration of the connector on the servo conveyor can be parameterised by the relative angle ϑ_1

The length from the main thrust bearing to the end connector is *L*. The configuration of the mobile platform is described by the position (X_p, Y_p) and the orientation Φ_P of frame {P} relative to some global frame {W}. The forward kinematics of the location of reference frame {C} with respect to the global reference frame {W} can be described as: (Abou-Samah *et al.* 2006).

$$\begin{bmatrix} X_C \\ Y_C \end{bmatrix} = \begin{bmatrix} X_P \\ Y_P \end{bmatrix} + \begin{bmatrix} \cos(\Phi_P) & -\sin(\Phi_P) \\ \sin(\Phi_P) & \cos(\Phi_P) \end{bmatrix} \times \begin{bmatrix} L\cos(\theta_1) \\ L\sin(\theta_1) \end{bmatrix}$$
$$\Phi_C = \Phi_P + \theta_1 \quad (1)$$

3.3 Trajectory Creation Mechanism

The required end connector trajectory, as specified by the motion requirements of the payload, is mapped into the platform {P} reference frame as a required cartesian pose as well as linear and angular velocities for the mobile platform. (Abou-Samah *et al.* 2006)

3.3.1 Trajectory generation via mapping of arbitrary end connector trajectory

By locating the base of the conveyor at the midpoint of the axle creates a coupling between the angle of the thrust bearing rotary housing and the orientation of the mobile platform. This configuration of the articulation allows for a direct mapping of end connector trajectory into required platform motions without any nonholonomic influence. (Abou-Samah *et al,* 2006)

$$\gamma = \Phi_P + \theta_1 \quad (2)$$

Equation (1) can now be written as:

$$\Phi_C = \gamma$$
$$\begin{bmatrix} X_C \\ Y_C \end{bmatrix} = \begin{bmatrix} X_P \\ Y_P \end{bmatrix} + \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) \\ \sin(\gamma) & \cos(\gamma) \end{bmatrix} \times L \quad (3)$$

Where, $L = \begin{bmatrix} L \\ 0 \end{bmatrix}$

Hence, given a desired end connector trajectory ($X_c^d(t), Y_c^d(t), \phi_c^d(t)$) , an expression for the required trajectory of the mobile platform can be determined as:

$$\gamma^r = \Phi_c^d$$
$$\begin{bmatrix} X_P^r \\ Y_P^r \end{bmatrix} = \begin{bmatrix} X_c^d \\ Y_c^d \end{bmatrix} - \begin{bmatrix} \cos(\gamma^d) & -\sin(\gamma^d) \\ \sin(\gamma^d) & \cos(\gamma^d) \end{bmatrix} \times L \quad (4)$$

At this stage the calculated $\gamma^r(t)$ can not be resolved into its required constituents, the orientation of the mobile platform $\Phi_P^r(t)$ and the orientation of the conveyor $\theta_1^r(t)$.

Differentiating (4) for constant L we obtain the desired Cartesian velocity of the mobile platform as:

$$\dot{\gamma}^r = \dot{\Phi}_c^d$$
$$\begin{bmatrix} \dot{X}_P^r \\ \dot{Y}_P^d \end{bmatrix} = \begin{bmatrix} \dot{X}_c^d \\ \dot{Y}_c^d \end{bmatrix} - \begin{bmatrix} -\sin(\gamma^d) & -\cos(\gamma^d) \\ \cos(\gamma^d) & -\sin(\gamma^d) \end{bmatrix} \times L \dot{\gamma}^d \quad (5)$$

Since the reference frame {P} is rigidly attached to the mobile platform with the *X*-axis in the direction of forward translation. This configuration permits the unique determination of both the required orientation of the mobile platform Φ_P^r and the magnitude of the desired forward velocity v_P^r. (Abou-Samah *et al. 2006).*

$$\Phi_P^r = \tan^{-1}\left[\frac{\dot{Y}_P^r}{\dot{X}_P^r}\right] \quad (6)$$

$$v_P^r = \sqrt{\left[\left(\dot{X}_P^r\right)^2 + \left(\dot{Y}_P^r\right)^2\right]} \quad (7)$$

Differentiating (6) provides the required angular velocity for the mobile platform.

$$\omega_P^r = \left[\left(\dot{X}_P^r \ddot{Y}_P^r - \dot{Y}_P^r \ddot{X}_P^r\right) / \left(\dot{X}_P^r\right)^2\right] \cos^2\left(\Phi_P^r\right) \quad (8)$$

From this information, controllers can be developed to converge the actual mobile platform pose onto the required platform pose, interested readers are referred to (Abou-Samah *et al.* 2006) for greater insight.

4. SOFTWARE ARCHITECTURE

4.1 Robot Servers and Generic Algorithm Development

In order to realise FMH systems the mobile platforms must communicate via well interpreted generic message passing primitives. To achieve this, a well known robot server namely Player, (Collett *et al*. 2006.), was used as a Hardware Abstraction Layer (HAL). The platforms developed here use the Vector Field Histogram +, (*Borenstein et* al. 1998), algorithm for local obstacle avoidance which was built into the server at compile time. Using the Player 2.0.4 API plug-in drivers were written to provide support for the control hardware used on the platforms. Four interfaces are supported on each platform via two plug-in drivers, Fig. 5. Fig. 4 shows a mobile platform with four device abstractions as well as the network addresses of the abstract devices.

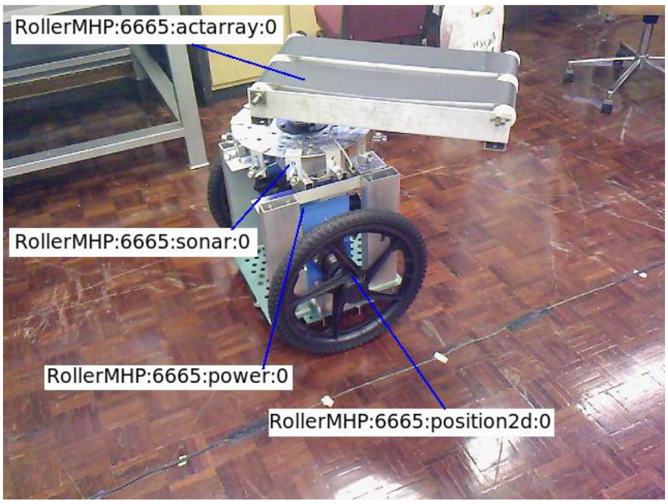

Fig.4: Network abstractions for the mobile materials handling platform.

The conveyor was abstracted as an array of linear and rotary actuators via the actuator array interface, (Biggs, G. MacDonald, B. 2006). This allows control of the conveyor system via message passing with generic actuators. The platforms drive system has been abstracted as a position2d device which can attain any configuration in SE(2). Other interfaces supported by the plug-in drivers are the power interface, which monitors battery voltage to enable a recharging routine as part of the task scope of the platform. A sonar interface makes the sonar data available to all interested parties. For a full set of interface specifications the reader is referred to (playerstage.sourceforge.net).

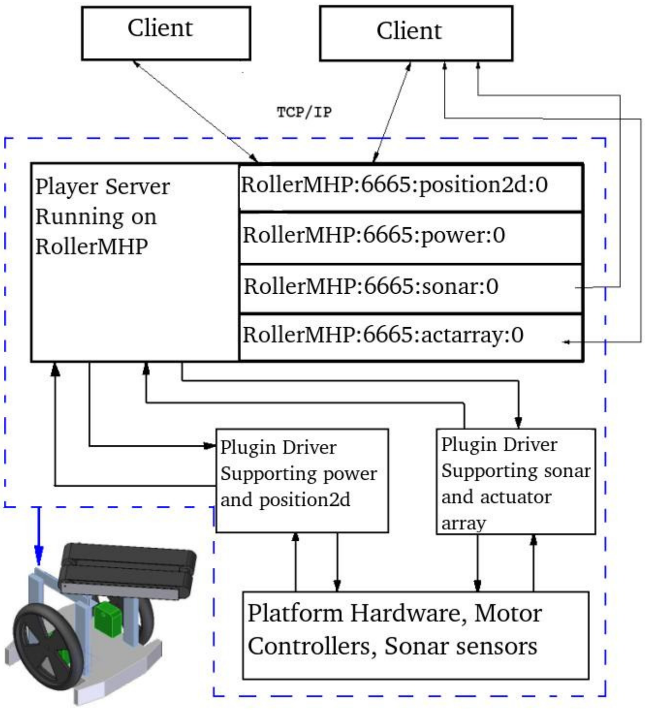

Fig. 5: Layout of an instance of the Player server running on one of the mobile materials handling platforms.

5. TASK SCHEDULING AND ALLOCATION SYSTEM

5.1 Operating System Theory Applied to FMH Task Scheduling Systems

Task scheduling and allocation is a requirement in the realisation of Flexible Materials Handling systems. The task scheduling and allocation system presented here builds on Operating System primitives. Many concepts used in UNIX and other Operating Systems can be applied to FMH systems. The concept of putting processes on sleep queues until operational conditions are met, i.e. blocking, can be used to develop mechanisms which put task instances, payload transportation, on sleep queues until destined machine tools or other infrastructure become available. Assembly machine tools can produce "critical sections" where initiation of assembly requires consecutive component delivery and material payload transportation tasks. The materials handling platforms should be able to perform the consecutive tasks whilst holding a semaphore and release the semaphore upon exiting the "critical section". The task scheduling and allocation of FMH systems should support standardised "system calls" and have plant specifics passed in as parameters. The conceptual outlay of the task scheduling and allocation system is depicted in Fig. 6.

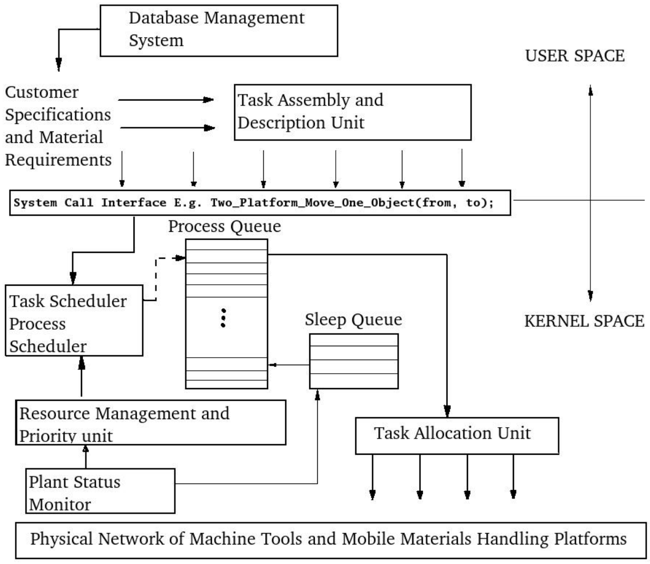

Fig. 6: Operating System analogy for FMH task scheduling and allocation system.

Separation of the task scheduling and allocation areas into "User Space" and "Kernel Space" helps define the environment characteristics of the system and provides contrast between the limited resources of the physical platforms and the unlimited resources of the virtual task assembly mechanisms and database systems. The task scheduling and allocation system is still currently under development and is an ongoing project. Mechanisms in the Robot Server to allow for "task dumps" in which the platforms receive subtask structures are currently under development.

6. DISCUSSION

The drive subsystem of the mobile platforms performed well during operation. On line tuning of the PID parameters provided smooth velocity control. The ability to configure the PID parameters during platform operation was beneficial and development is under way in producing a control loop tuning system which automatically configures the PID parameters based on the effective material loads seen by the mobile platforms. This will aid in allowing the platforms to retain optimal motion control as their physical bandwidths change with payload.

The conveyor alignment accuracy was tested for repeatability and performed well. This is a requirement if the platforms are to transfer materials between one another in a robust manner. The system works by utilising the signal from the digital compass to provide global conveyor orientation. When the platforms are in conveyor range the receiving platform then calculates the error between its conveyor orientation and the orientation of the conveyor that will transfer the payload. The error signal is used to drive the conveyor to the correct orientation. Once the payload has been successfully transferred both platforms update their orientation estimates by comparing their global and local orientation estimates. The local orientation is derived from the dead reckoning estimate of platform pose and the readings from the quadrature encoders fitted at the two articulations on the conveyor.

The Player robot server used in the research provided usable interface specifications without any modifications. The new Application Programming Interface (API) of the 2.0.x versions of the server allowed for the simple and efficient development of plug-in drivers achieving quick support of the hardware devices used by the mobile platforms. The Vector Field Histogram obstacle avoidance algorithm provided good local obstacle avoidance for the mobile platforms. Although the platforms designed in this paper are homogeneous in nature in the sense that they all have the same hardware and capabilities, this is not a requirement of such materials handling systems. Furthermore multi-agent cooperation and coordination algorithms parameterised by control bandwidths built using Players interfaces could allow for well behaved cooperation between heterogeneous materials handling platforms.

The conceptual design of the task scheduling and allocation system lends itself to the development of universal standards in FMH system calls. The task assembly mechanisms, which map customer specifications and material requirements to task instances can rest on well defined operations and allow international development of system call interfaces. The system also explicitly integrates the status mechanisms of the machine tools into the resource management system allowing for more diffuse material transfer through the manufacturing system by catering for different machine cycle times.

7. CONCLUSION

In this paper a mobile platform architecture was developed which can operate effectively in singular context as well as during cooperative tasks involving multiple mobile platforms.

A Robot Server was utilised in providing the platforms with generic network interfaces and message passing primitives.

The conceptual layout of a task scheduling and allocation system was presented and based on Operating System analogies.

REFERENCES

Abou-Samah, M., Tang, C. P., Bhatt, R. M and Krovi, V (2006). A kinematically compatible framework for cooperative payload transport by nonholonomic mobile manipulators. In: *Autonomous Robots*.

Abou-Samah, M and Krovi, V (2001). Cooperative Frameworks for Multiple Mobile Robots. In: *CCTOMM Symposium on Mechanisms, Machines and Mechatronics.*

Bhatt, R., Tang, C. P. and Krovi, V (2004a). Geometric Motion Planning and Formation Optimization for a fleet of Nonholonomic Wheeled Mobile Robots. In: *Proceedings of the 2004 IEEE International Conference on Robotics and Automation,* pp 3276-3281, New Orleans, LA.

Bhatt, R., Tang, C. P. and Krovi, V (2004b). Decentralized Kinematic Control of Payload Transport by a System of Mobile Manipulators. In: *Proceedings of the 2004 IEEE International Conference on Robotics and Automation,* pp 2462-2467, New Orleans, LA.

Biggs, G., MacDonald, B (2006). Generic interfaces for robotic limbs. In: *Proceedings of the Australasian Conference on Robotics and Automation.* Auckland, New Zealand.

Borenstein, J. and Ulrich, I (1998). VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots. In: *Proceedings of the 1998 IEEE International Conference on Robotics and Automation.* pp. 1572-1577. Leuven, Belgium.

Bostelman, R., Hong, T., Chang, T., Shackleford , W. and Shneier, M (2006). Unstructured Facility Navigation by Applying the NIST 4D/RCS Architecture. In: *Proceedings of the International Conference on Cybernetics and Information Technologies, Systems and Applications,* Orlando Florida.

Collett, T., MacDonald, B. and Gerkey, B (2005). Player 2.0 Toward a practical robot programming framework. In: *Proceedings of the Australasian Conference on Robotics and Automation,* University of New South Wales, Sydney, Australia.

MMHS Metamorphic Materials Handling System Final Report, Doc No. IMS/ISC/11/6 – 3/1 Issue:1., 2003.