3D Simulation of Mobile Manipulator Based on Virtual Prototyping Technology

Yang Qian [1, a], Ahmed Rahmani [2, b], Qian Chen [1, c], Songhua Shi [1, d], Fei Han [1, e]

[1] Shanghai Radio Equipment Research Institute, Shanghai, 200090, China,

[2] LAGIS, FRE CNRS 3303, Ecole Centrale de Lille, Villeneuve d'Ascq, 59650, France

[a]qianyang87@163.com, [b]ahmed.rahmani@ec-lille.fr, [c]qian_ch@163.com, [d]snake7946@sina.com, [e]mffthan@126.com

Abstract

This paper presents the design and 3D simulation of a mobile manipulator used for domestic tasks. In order to make the mobile manipulator's design more efficient, a virtual simulation system is built using virtual simulation technology. First, a 3D model of the mobile manipulator is developed in Solidworks, the model is then imported into the dynamic simulation software Adams for simulation. Finally, through the interface modular part of MATLAB and Adams, a mobile manipulator control system is established with Simulink to successfully implement a coordinated simulation.

Keywords

Mobile manipulator; coordinated simulation; virtual prototype.

1. Introduction

Mobile manipulator is nowadays a widespread term which attracted significant attention over the years[1]-[6]. It generally consists of one or two robotic arms and a mobile platform. The objective of the undertaken project is the development of a personal assistant robot prototype for assistance tasks in household environments.

Mobile manipulator is a complex electromechanical system. The traditional design methods (prototype repeated testing and amelioration) are not only difficult to effectively improve the robot's performance, but also spend a lot of material and time. Simulation is an important and useful technique that can help users to better model and understand real life systems. Virtual prototyping is now a very common technique to simulate mechanical systems. Using software tools, one can design and model systems by simulating separate parts of these systems and visually investigating their behaviors under conditions that are close to real situations. Multi-Body Dynamic Simulation package MSC. Adams (Automatic Dynamic Analysis of Mechanical Systems) developed for 3D motion simulation can facilitate the simulation of the mobile manipulator's kinematics and dynamics. Its motion simulator combines 3D mechanical drawings of different subsystems into a complex mechatronic system. This eliminates the need for deriving complex, nonlinear mathematical model for the dynamic behavior of the system. Adams is integrated with MATLAB/SIMULINK so that it is capable of creating a control-centric design platform. Through interactive coordinated simulation, many technical bases can be provided for the research of the actual physical prototype, thus the performance of the robot can be effectively improved. Use of virtual prototyping for design and development of a humanoid robot has been reported in a recently published article[7].

This paper presents the definition, design, and simulation of a mobile manipulator in a virtual environment that integrates Adams with MATLAB. Section 2 gives the application design for the personal assistant robot. And the requirements for realizing those applications are discussed. The mechanical structure and its drawings are briefly described in section 3. The system design process' details are given in section 4. Section 5 presents the simulation results with the controller implemented in MATLAB/Simulink. Finally, section 6 gives some concluding remarks.

2. Applications and Requirements

2.1 Applications

Christensen[8] named three groups of domestic applications (entertainment, assistance for elderly and handicapped, everyday tasks) for robots. Manja Lohse proposed six categories (entertainment, personal assistant, healthcare/care giving, companionship, toy, Pet)[9]. Fong et al. also differentiated between a toy and an entertainment robot [10].

In the functional design process we analyze the different classification methods and design the following applications.

Healthcare /Care giving:

- Communication and social integration
- Technical house management (Infrastructure)
- Fetch and carry tasks
- Mobility support
- Personal security
- Personal management

Personal Assistant:

This category includes robots for cleaning and other household chores.

- Meal preparation and serving (e.g. micro wave, stove)
- Pouring out drinks
- Setting tables
- Loading and unloading dishwashers
- Simple cleaning and tidying tasks (sweeping a floor by using a bloom, gathering clothes and putting it in a washer etc)
- Vacuum cleaning

Companionship and Entertainment:

- Tracking and following
- Facial expressions
- Playing music
- Providing information (e.g. weather)

In addition to these domestic functions, our robot also can work in public places as a **public assistant**:

- Entertainer: it approaches the visitors and welcomes them
- Keeping tracking of people
- Guide: it gives guided tours
- Transport of objects
- Security patrolling

2.2 Requirements

For realizing these robot applications, some specific requirements have to be fulfilled to maximize the usefulness of the assistant robot. These include the following:

- Mobile Base

A personal assistant robot imposes high requirement on the employed mobility concept. Usually the robot performs its tasks in environments where humans can walk around without any problems. Most of those environments have in common that wheeled vehicles can access them.

- Manipulation System

According to the anthropomorphic model two dexterous hands mounted on a pair of humanoid robotic arms should be used for the manipulation system in order to manage some sophisticated tasks, e.g., opening bottles. To manipulate arbitrarily positioned and oriented objects the manipulation system needs at least six DOFs. In addition, a bendable body is necessary to help the robot pick up objects from the ground.

- Sensor System

Sensors are used to give feedback of both interior and exterior states of the robotic system and to supervise the current interaction of its actuators with the environment. Similar to humans the robot's sensor system can be subdivided into exteroceptors and proprioceptors.

- Information Processing and Robot Control

A complex robotic system depends on fast and reliable sensor data processing as a basis for intelligent robot control. Usually a control board is used to capture image, process image, make decision, and control motions for the robot. A hierarchically structured multi-processor system seems to be the best solution. Different types of processors are to be chosen for every hierarchical level.

- Man-Machine Interface

Research in this area focuses on natural human channels of communication, such as graphical user interface, touch user interface, natural language and gestures, to provide easy and friendly multimodal interfaces between humans and the robot.

- Safety Precautions

For establishing a system which can provide the robot with abilities to achieve some tasks, failure detection and recovery abilities are necessary. On the other hand, the robot working in close proximity to humans, even sharing their operating envelopes, is exposed to a special danger. Safety can be enhanced by placing tactile sensors on the robot's surface to ensure that the robot could not harm humans.

- Actuators

Actuator provides the driving force and motion for robot joint. Given the wide variety of existing actuators, the selection is based on matching the performance characteristics of the actuator, such as force and displacement, to the requirements of the given tasks.

- Power Supply

A power supply can put out a reasonable amount of current for servos and other small motors as well as run a processor and sensors. For a completely autonomous robot, a sensor sensing the battery charge should be added. If the charge goes below a certain limit, the robot should stop the task and look for a charging station without any human intervention.

- Cost

Two competing factors are considered when designing the robot: cost and functionality. The assistant robot needs enough functionality to perform various tasks. While at the same time, it must be simple enough to keep the cost low. Low cost can be achieved by (a) using readily available components, (b) designing a reasonable control system to reduce high computational power which results in high cost of the equipment and even higher cost for developing the software for the control applications and (c) reducing the complexity of mechanical design.

3. Mechanical Design

3.1 Mechanical Configuration

The mechanical design of the complete robot requires careful and complex tradeoffs between form, function, power, weight, cost and manufacturability. Our assistant robot consists of a mobile platform and a dual-arm/hand system with adequate payload and dexterity characteristics. For example, the mobile base should be able to transit doors and navigate in human-sized environments. The fingertips of the dexterous hands should reach a reasonable fraction of the workspace reachable by a standing person and possibly be able to pick up objects on the ground.

For realizing the capabilities mentioned above, our human-like mobile manipulator prototype (Fig. 1) roughly stands 140 cm tall and weighs 80 kg. The high number of DOFs allows the robot to realize complex motions. Fig. 2 describes the structure of the mobile platform. Fig. 3 shows the structure of the robotic hand.

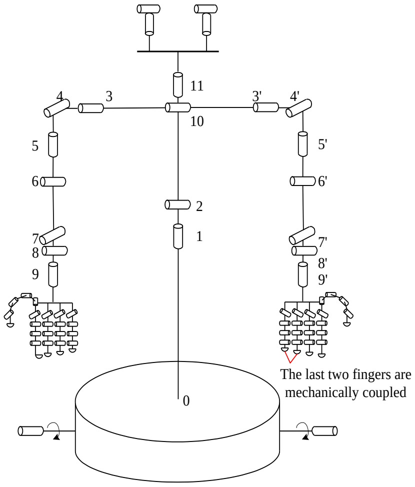


Fig. 1 Dimensions of robot model

Table 1. DOFs configuration

Part	DOF
Eyes	2 (pan-tilt) ×2
Neck	2
Waist	2
Arms	7×2
Hands	16×2
Mobile base	2
Total	56

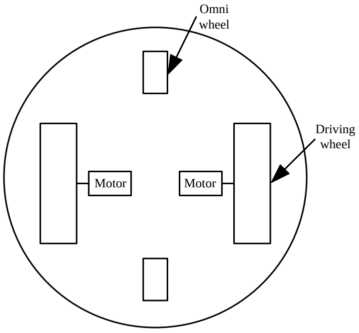


Fig. 2 Structure of mobile platform

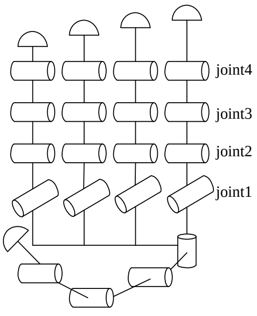


Fig. 3 Structure of dexterous robotic hand

From the description of the dimensions, it can be seen that the robot's head can not only rotate left and right, but also rotate from up and down. It also can independently move the eyeballs because the robot has 2 DOFs for each eye (for a camera pan and tilt) which allows a wide field of vision change. As a consequence, the robot's head can obtain all needed information. Our robot is equipped with two 7 DOFs arms like human arms, and with two five-fingered hands. The shoulder gives pitch, yaw and roll, the elbow allows for pitch, and the wrist allows for pitch, yaw and roll. The upper body also consists of a 2 DOFs waist. This bendable waist allows the robot to be able to pick up objects on the ground. The robotic dexterous hand has a similar configuration with the human hand. It has five fingers and each finger is composed of 4 joints. The thumb and the other four fingers (index, middle, ring and little fingers) have similar structure in terms of kinematics and dynamics. The last two fingers (the ring and the little fingers) are mechanically coupled so that they keep the same motion, therefore together they just have 4 DOFs and the little finger only plays a supportive role. From this description the dexterous hand has 16 DOFs in total. This design can achieve the dexterity of human hand such that it can be used to perform various operations.

Most commercial manipulators are heavy, large, and power hungry. Highly mobile robots, such as legged mobility platforms, cannot easily accommodate dexterous manipulators. To overcome this difficulty, a differential drive mobile platform with two conventional wheels and two supporting wheels has been chosen. The wheeled drive is chosen over leg drive because of safety and stability during manipulation.

Normally, a DC servomotor drives each joint with a rotary encoder. The mobile platform includes a battery pack, a laser scanner and a processor for navigation tasks. The robot is equipped with a force sensor in each arm and two color CCD cameras on its head. Each dexterous robotic hand needs as a minimum a set of multi-axis force/torque sensors and tactile sensors in all fingertips and position sensors at all joints to enable control schemes like force/torque control and position control in autonomous operation and teleoperation.

In the actuator system, robotic hands usually can be divided into two types: built-in actuator type and external actuator type. Built-in actuator type of robotic hands generates the motion of fingers by using motors installed inside the finger or palm. This type of robotic hand drives the joint by direct drive or gear drive. External actuator type makes the structures of its fingers simple and light by using wire or belt driving mechanism. In our project, we choose the latter one. The fingers which are to be actuated independently through motors placed at remote location.

Based on the above mechanical configuration, compared to the traditional industrial mobile manipulators, this robot is capable of performing various tasks in unstructured or semi-structured environments, thus it could address greater challenges in autonomous localization and navigation, object identification, control and coordination.

3.2 Joint Angle Range

In order to ensure the robotic model can move similarly to a human, the angle ranges for the robot are defined according to the following two considerations:

- The specific angle ranges of human being

In a human body, each joint has its own reachable angle limit. Referring to the material[11], the angle ranges of this humanoid model are defined.

- The maximum rotated angle

The other point to be concerned about is the maximum rotation angle of servo motors used to build up the real model is only 180 degree.

Table 2 and Table 3 make a comparison between the joint angle ranges of the human and the humanoid robot. These movable ranges of the robot are similar or over the human's ones.

Table 2. Comparison between human and humanoid robot angle ranges

Joint	Human (deg)				Robot (deg)			
	number	yaw	pitch	roll	number	yaw	pitch	roll
Neck	3(yaw, pitch, roll)	-80-80	-45-45	-45-45	2(yaw, pitch)	-90-90	-45-45	
Waist	3(yaw, pitch, roll)	-30-30	-30-90	-30-30	2(yaw, pitch)	-90-90	-10-120	
Right/left shoulder	3(yaw, pitch, roll)	-90-50	-90-40	-135-40	3(yaw, pitch, roll)	-90-90	-180-90	-135-10
Right/left elbow	1 (pitch)		-10-150		1(pitch)		-10-150	
Right/left wrist	3(yaw, pitch, roll)	-30-40	-70-70	-90-90	3(yaw, pitch, roll)	-90-90	-90-90	-90-90

Table 3. Angle ranges of finger joints

Joint	Human (deg)	Robotic finger (deg)
Joint1	-15-15	-90-90
Joint2	0-90	-30-120
Joint3	0-110	-30-120
Joint4	0-90	-30-120

4. Virtual Prototyping

4.1 Solid Model Creation

In this paper, Solidworks is used to build a solid model for the assistant robot (Fig. 4). The obtained 3D model is imported into Adams software directly.

Each part's size and shape of this robot are created as those of the actual component. To reduce the complexity of the model in Adams, only the mechanical parts are included in the Solidworks drawing. This structure is sufficient to reflect the physical properties of the mobile manipulator, which will be examined later.

4.2 Virtual Prototype Development

MSC. Adams is one of many computer programs for modeling and simulating multibody systems. In Adams a virtual prototype of a mechanism can be built and then simulated. And it also allows users to import part geometry from CAD programs. The most popular file format to import into Adams is Parasolid (*.x_t) which increases the chances of receiving solids when importing geometry. Using Parasolid is convenient because one file contains all the geometry and Adams /Exchange creates a separate part for each solid. After the parts of the mechanical system are developed with Solidworks, we import the solid model of the mechanism which has already been saved as *.x_t document. All the constructional and technological features specified at the designing stage are represented in the mechanical virtual model. Fig. 5 shows the virtual prototyping model in Adams.

The elaborated Adams model takes several aspects into account, such as joint constraints, mass, inertial properties, contact constraints, friction, actuation forces and reference markers, which enable good approximation of the real robot behavior. First, each joint of the mechanism is defined with a particular motion. For example, the driving joint is specified as a revolute joint so the rotation associated with this joint specifies the rolling motion of the wheel. Secondly, according to the design parameters of the robot, material, mass and moment of inertia tensor should be defined in Adams. Adams calculates the mass and inertia of each rigid body based on its solid geometry and its material type. The default properties for the part can be modified by entering the properties we need. For each part a body-fixed reference frame with its origin at the center of mass is defined in Adams as a marker. Additionally other reference markers could be created if needed. Table 4 lists the main dynamic characteristics of the robot components for the proposed Adams model (excluding the hands). The symbols Ixx, Iyy and Izz are frequently used to express the moments of inertia of each link about its three axes passing through the link's center of mass. The products of inertia are given by Ixy, Iyz and Izx. Contact constraints and friction will be discussed in the next section.

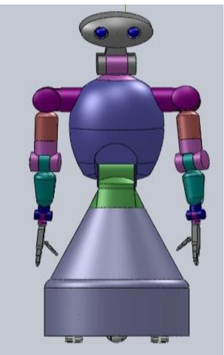


Fig. 4 Solid works drawing of mobile manipulator

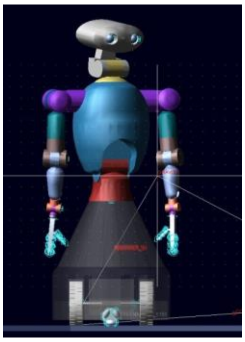


Fig. 5 Virtual prototyping model in MSC. Adams

Table 4. Main inertial properties of robot

Part	Mass(kg)	Ixx(kg mm2)	Iyy(kg mm2)	Izz(kg mm2)	Products of inertia around center-of-mass (kg mm2)		
					XY	YZ	ZX
Head	2	1.452×104	1.196×104	1.177×104	0	0	0
Neck	2	3180.97	3020.63	2687.458	0	0	0
Upper body	10	1.608×105	2.419×105	2.096×105	0	0	0
Waist	6	2.112×104	2.118×104	1.87×104	0	0	0
Lower body	50	1.796×107	1.961×107	2.83×107	0	0	0
Arm1	2.477	9152.28	6142.9	6363.91	0	0	0
Arm2	1.192	4041.89	1086.44	4465.585	0	0	0
Arm3	0.57	1436.32	1133.86	939.69	0	0	0
Arm4	0.898	607.69	3698.59	3851.59	0	0	0
Wrist1	0.273	103.248	163.444	218.373	0	0	0
Wrist2	0.096	26.087	11.966	30.698	0	0	0
Wrist3(with gripper)	0.3	657.051	792.638	1326.87	0	0	0

Dexterous hand is a complex mechanism with multiple-degree-of-freedom. It is an exciting field with still many unsolved problems in both mechanical design and control. It is necessary to create a separate

3D model of the robotic hand for research. Table 5 lists the main dynamic characteristics of the robotic fingers.

Table 5. Summary of mass and inertia characteristics of finger

Part	Mass (kg)	Rotation inertia around center-of-mass (kg·mm2)			Products of inertia around center-of-mass (kg·mm2)		
		Ixx	Iyy	Izz	Ixy	Iyz	Izx
1	0.0024	0.127	0.149	0.134	0	0	0
2	0.005	0.301	0.646	0.698	0	0	0
3	0.005	0.301	0.646	0.698	0	0	0
4	0.01339	0.5466	0.681	0.717	0	0	0

Additionally, a numerical integrator is used and should be chosen for the simulation. The integrator is an algorithm that solves the differential equations of a dynamic problem over an interval of time during a simulation. The selected integrator for simulation is the Gear Stiff integrator (GSTIFF). The GSTIFF integrator is the default in Adams environment since it provides good solutions for simulation of stiff mechanical models.

One has to specify simulation frequency, internal frequency and step size. Simulation frequency is the frequency of updating the graphic display, whereas internal frequency is closely related to the system under study. It represents the speed at which component states are changed. Internal frequency is a very important parameter in simulation setting and other solver settings must be set in accordance with the internal frequency.

Once the virtual prototype is complete, Adams checks the model and then runs simultaneous equations for kinematic, static, quasi-static, and dynamic simulations. The results are viewable as graph, data plots, reports, or colorful animations.

As a test to verify the reasonableness of the design, the right and left dexterous hands are modeled respectively. The simulation test in Adams shows the effectiveness of the developed mechanism in grasping objects of various shape and size. Fig. 6 demonstrates the virtual prototyping model of right hand in Adams. Fig. 6(a) shows the initial configuration of the hand-object system. Fig. 6(b) shows the system configuration when the hand is grasping the object.

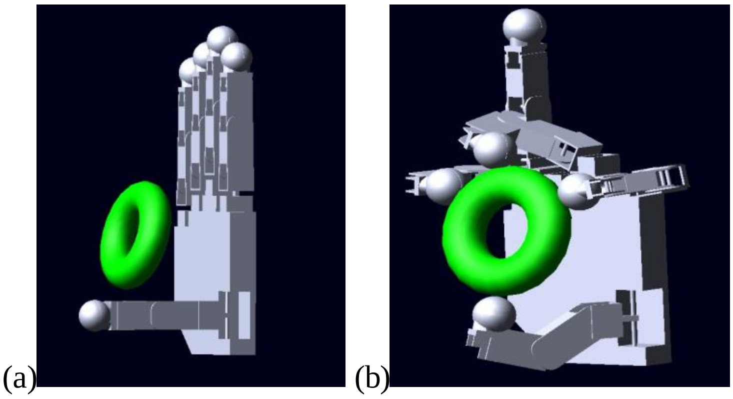


Fig. 6 Virtual prototyping model of dexterous hand in MSC. Adams (a) Initial configuration of hand-object system (b) Final configuration

4.3 Contact Modeling between Bodies in Adams

In this section we will present how the contacts are modeled in Adams. The connection between each wheel and the floor is modeled using a contact with impact function. The parameters of hardness of impact force, force exponent, damping, penetration depth and friction, etc., must be set. If the contact parameters are incorrectly specified, we will create a system which requires a lot of computational time to solve. That would be in case we would choose high stiffness and small damping. The system would oscillate on high frequency and would be difficult to solve. With the right parameters, the model behaves like a soft tire.

The constraints between the fingers and the object are also contacts with impact functions. Then the contact constraints need to be created between each finger and the object, and the parameters also must be set.

The impact function is written out:

$$F_n = k\delta^e + step(\delta, 0, 0, d, C)\dot{\delta} \tag{1}$$

Where F_n is the contact force, k denotes the stiffness, e represents the force exponent, δ is the distance between the contacting bodies, C is the damping coefficient and d denotes the penetration depth.

Eq. (1) shows that the contact force defined in Adams is composed of two parts. An elastic component $k\delta^e$ acts like a nonlinear spring. The other one $step(\delta, 0, 0, d, C)\dot{\delta}$ is the damping force, which is a function of the contact-collision velocity.

The coulomb friction, F_f, is calculated by multiplying a friction coefficient, μ_f, with the contact force, F_n .

$$F_f = \mu_f F_n \tag{2}$$

The friction coefficients chosen are 0.95 for the static condition and 0.8 for the dynamic condition.

4.4 Co-Simulation Connection between Adams and MATLAB/Simulink

After the mechanical part has been built, a co-simulation platform is set up to develop a control system for the complete robotic system. Adams has toolbox Adams /Controls that will prepare our model for connection with other software designed for control. One can choose between MATLAB/Simulink and MSC.Easy5. In this paper co-simulation with MATLAB/Simulink is used. The detailed procedure is shown in Fig. 7 as a flowchart. First, a trajectory planning system is built by Simulink. The feasibility of the motion generated in the first step is then verified through the collaborative simulation between Simulink and Adams. The feasible results will be used to the physical prototype.

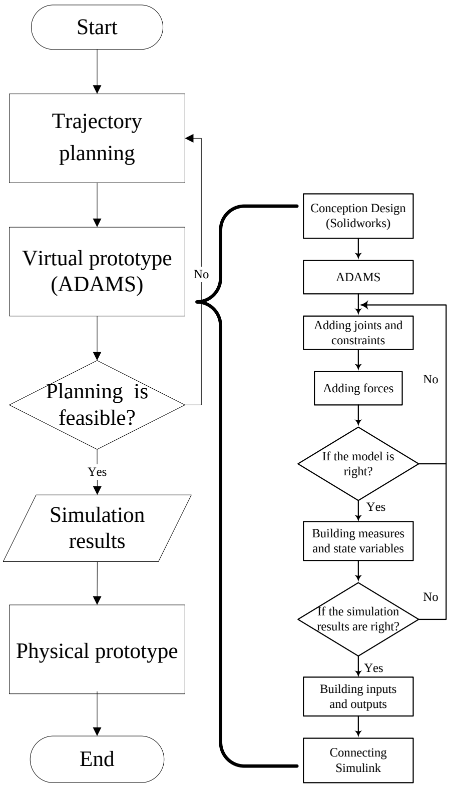

Start
Trajectory planning
Virtual prototype (ADAMS)
No
Planning is feasible?
Yes
Simulation results
Physical prototype
End
Conception Design (Solidworks)
ADAMS
Adding joints and constraints
Adding forces
No
If the model is right?
Yes
Building measures and state variables
No
If the simulation results are right?
Yes
Building inputs and outputs
Connecting Simulink

Fig. 7 Flowchart of collaborative simulation

Below is a more detailed explanation of the procedure. First, the model built in Solidworks is imported into Adams. Then the joint constrains and the properties are added. Next, the measures and the state variables should be built. Simulations are created in Adams to verify the virtual prototype. The correctness of Adams simulation results can be verified by comparison with the theoretical results.

After defining the input and output variables, it can form a closed information loop between Adams and MATLAB. MATLAB reads the output information of the plant and calculates the control output while the Adams solver solves system equations for the plant. Adams /Controls and MATLAB communicate by passing the state variables back and forth (Fig. 8). Our main goal is to control the joints to make the robot perform some human tasks. To achieve this goal, the joint torques are defined as the input variables; the joint angles and joint angular velocities are defined as the output state variables.

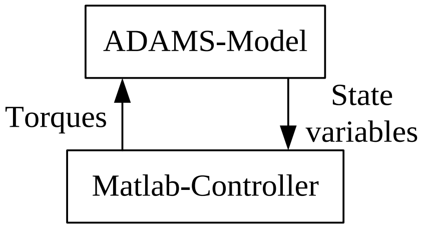


Fig. 8 Information flowchart of collaborative simulation

The various steps involved in setting up a coordinated simulation between Adams and Simulink are:

- Loading Adams /Controls
- Defining Input and Output Variables
- Referencing Input Variables in the Adams Model
- Exporting the Adams Block
- Connecting the Adams Block and the CES Block in Simulink
- Running the Co-simulation

Note that the output step size will determine the communication interval between Simulink and Adams and must be the same.

Through the co-simulation we can validate the virtual prototype of the personal assistant robot. Then based on the virtual prototype, the efficiency of some complex motion planning algorithms and control schemes can be verified.

5. Simulation Results

In order to validate the Adams model of the personal assistant robot some simulations are carried out in this section. The simulation results can verify the reasonableness of the mechanical design. This validation will result in the simulated behavior that is closer to the reality.

5.1 Workspace Computation

Exact computation of the workspace of robot manipulator is very important for its optimum design and application. The workspace of robot manipulator is defined as the set of points that can be reached by its end-effector and dependent on the joint angle range. The joint angle range of our robot has been defined in the mechanical design process and then the workspace can be computed. Fig. 9 (a) to Fig. 9 (d) show the workspace of a single arm including the motion of waist when the mobile platform remains stationary from different views. Fig. 9 (e) and Fig. 9 (f) show the robot workspace of the dual-arm/waist system. The shape and volume of the workspace, which we most concerned, is illustrated. This cartesian workspace forms a kind of capability map for the redundant arms of our robot. For example, the information can be used to search for valid grasps in regions where the hand can easily be placed by the robotic arm.

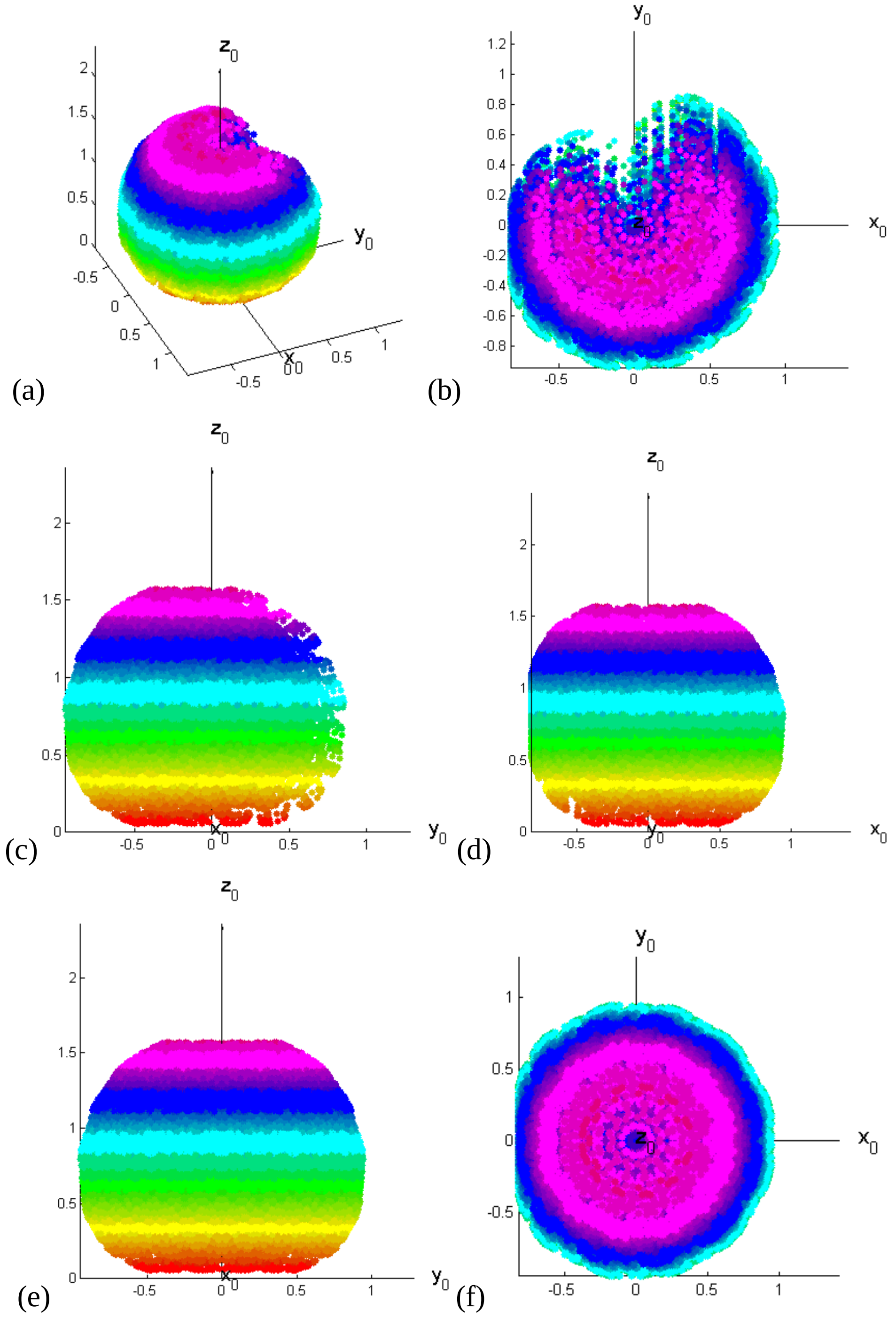


Fig. 9 3D workspace computed by MATLAB (a) 3D view (b) x-y view (c) y-z view (d) x-z view (e) y-z view with two arms (f) x-y view with two arms

5.2 Adams Simulation

The virtual prototype with the Adams View program is designed the same as the real mobile manipulator. As Adams is meant for 3D simulation of dynamics, it has very limited capability for realizing a controller. We create an interface between Adams and MATLAB/Simulink so that the controller can be implemented using Simulink. To call the generated plant in MATLAB, the plant name should be entered in the “command window” so that the input and output information will appear. By executing “Adams_sys” command, the blocks containing information about dynamic model of robot in Simulink environment are loaded (Fig. 10). By having “Adams-sub” block in Simulink environment, the robot model is suitable for control and motion simulation as a defined system in MATLAB.

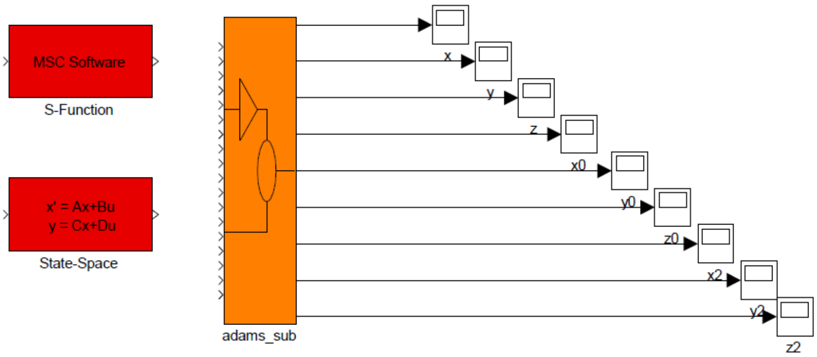


Fig. 10 Model plant and state space blocks

A velocity level controller shown in Fig. 11 is used as a closed-loop feedback controller and developed in Simulink. The general inverse solution of a kinematically redundant robot at the velocity level has the following form:

$$\dot{q} = J^{\dagger}(\dot{x}_d + K_p(x_d - x)) + (I - J^{\dagger}J)\dot{\varphi} \tag{3}$$

where K_p is a symmetric positive definite matrix, and the choice of K_p guarantees that the error uniformly converges to zero; x represents the end-effector position and x_d is the desired end-effector position; J is the Jacobian matrix of the mobile manipulator and $J^{\dagger}$ is its pseudoinverse. A typical choice of the null space joint velocity vector is[12]:

$$\dot{\varphi} = k_w\left(\frac{\partial w(q)}{\partial q}\right)^T \tag{4}$$

with $k_w > 0$. q Represents the vector of joint variables, $w(q)$ is a scalar objective function and $(\partial w(q)/\partial q)^T$ is the vector function representing the gradient of w. In this research, just the joint limit avoidance is considered as local optimization of a performance criterion. Thus the redundancy may be exploited to keep the manipulator off the joint limits.

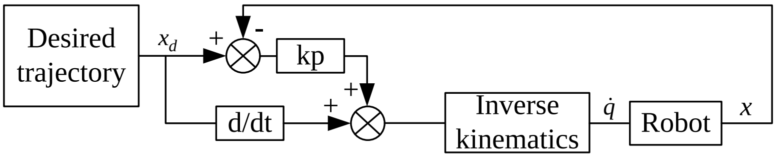


Fig. 11 Redundancy resolution at the velocity level

Fig. 12 demonstrates the block diagram of closed-loop control in Simulink and through it the co-simulation between MATLAB and Adams can be achieved.

As shown in Fig. 12, Adams outputs the end-effector positions of left and right arms respectively. The desired trajectories of the end-effectors have been given as a constraint. According to this condition, the position errors can be acquired. And through the controller, Simulink gives each joint's angular velocity to reduce the position errors.

First, suppose that the fingers, the right arm and the mobile platform remain stationary. The left manipulator is tested with a given trajectory based on the Jacobian pseudoinverse method. Fig. 13 (a) draws the desired trajectories along x, y, and z-axis. The reference in task space is generated by the equations:

$x = x_{l0} + 0.01t, y = y_{l0} + 0.015t, z = z_{l0} + 0.02t$.

Where $[x_{l0} \quad y_{l0} \quad z_{l0}]^T$ is the initial position of the left end-effector? The end-effector moves with a reasonable speed $[0.01 \quad 0.015 \quad 0.02]^T$ along three orthogonal directions respectively. Fig. 13 (b) depicts the motions in these three directions. Fig. 13 (c) shows the practical velocities of the end-effector and Fig. 13 (d) shows the trajectory tracking errors. Comparing Fig. 13 (b) with Fig. 13 (a), we can see that the manipulator can move following the desired trajectories. Fig. 14 gives the robot's final configuration after execution of the task.

Secondly, we validate the robot’s kinematics without hands (mobile base + dual manipulators). The mobile platform and the right manipulator, which compose Loop a, are regarded as a redundant system and the motion control method Eq. (3) is used. The reference trajectory in task space is given to the right end-effector. Each joint of the left arm keeps the same motion with the corresponding joint of the right arm. The reference in task space is generated by the equations:

$$x = x_{r0} + 0.01t, y = y_{r0} + 0.015t, z = z_{r0} + 0.02t$$

where $[x_{r0} \quad y_{r0} \quad z_{r0}]^T$ is the initial position of the right end-effector. The right end-effector also moves with a speed $[0.01 \quad 0.015 \quad 0.02]^T$ along three orthogonal directions respectively. The difference between this task and the previous task is the motions of the other arm and the mobile base.

The simulation results are shown in Fig. 15. Fig. 15 (a) and Fig. 15 (b) depict the desired trajectories, the practical motions in the x, y and z directions, respectively. Fig. 15 (c) shows the trajectory tracking errors. Fig. 15 (d) gives the motions of the mobile platform along x, y directions and the heading angle. The robot’s final configuration is shown in Fig. 16. Comparing the results of two simulations we find that the curves in the x and y directions oscillate more obviously. That result indicates that adding the mobile part seriously affects the dynamic behavior. In other words the mobile manipulator has a complex dynamic model, strong coupling and no holonomic constraints which are possibly caused by the mobile platform. Next step, we will take into account the dynamic characteristics when designing the control algorithm.

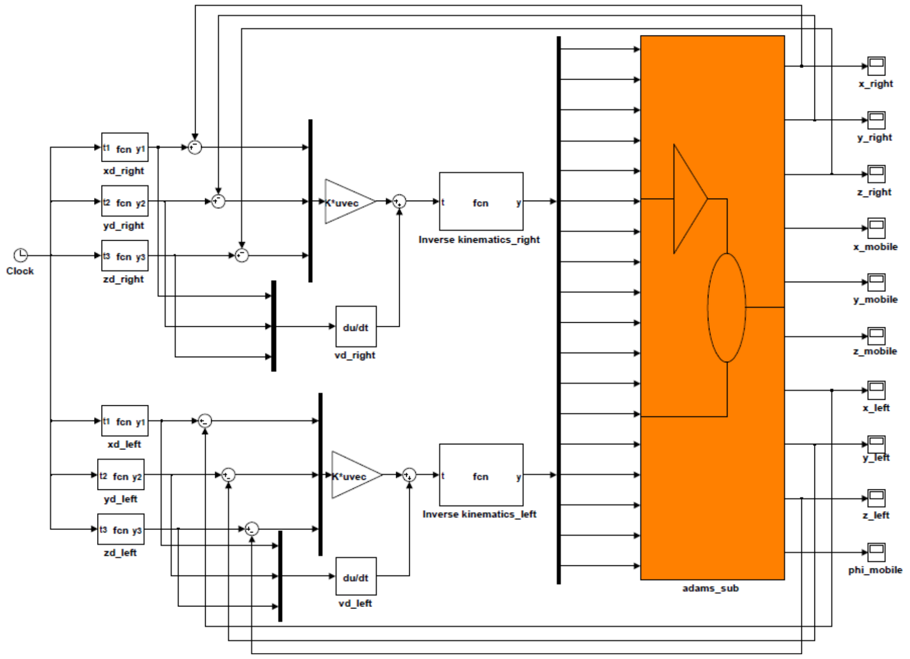
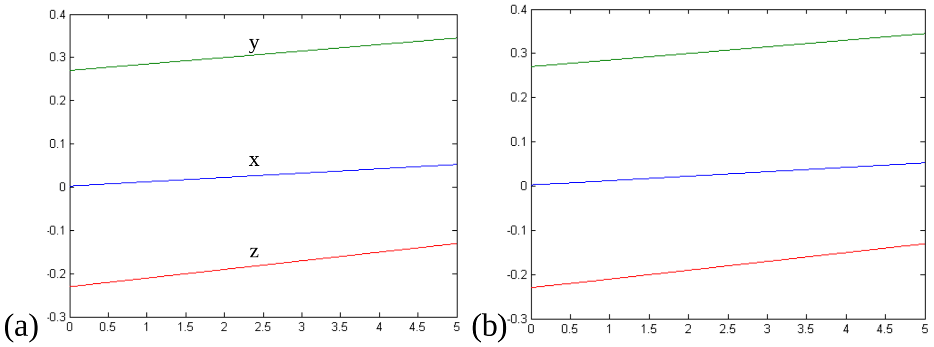


Fig. 12 Block diagram of closed-loop control in Simulink



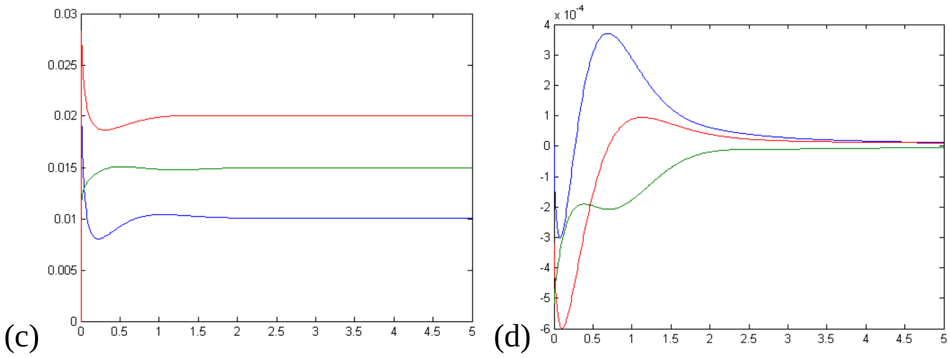


Fig. 13 Simulation results of one redundant manipulator (a) desired trajectories (b) Practical trajectories (c) Practical velocities (d) Trajectory tracking errors

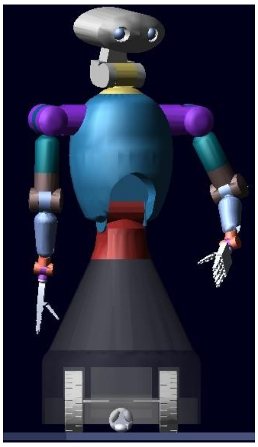


Fig. 14 Final configuration

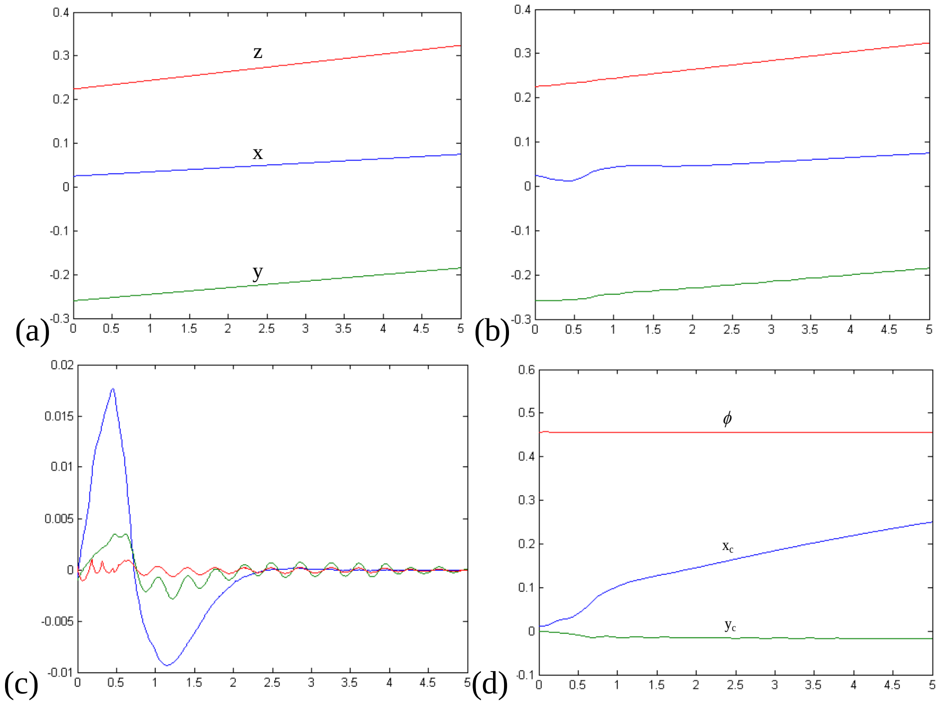


Fig. 15. Kinematic simulation results of complete mobile manipulator (a) Desired trajectories (b) Practical motions (c) Trajectory tracking errors (d) Motions of mobile platform

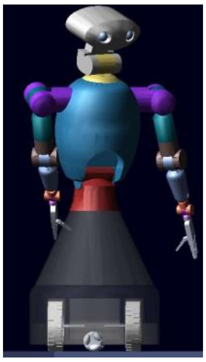


Fig. 16. Final configuration

6. Conclusion

Several tests have been carried out to verify the principle of the robot's operations. A controller which combines Adams with MATLAB / Simulink is implemented to strengthen the justification of building such virtual robot. The simulation results prove that the method is correct and efficient. Therefore, the virtual robot created in Adams represents the actual mechanical system. Based on the simulation results, one can perform design changes to improve the robot performances. The virtual prototype can now be used for various investigations, which are otherwise time-consuming or costly.

References

[1] Y. Yamamoto and X. Yun, A modular approach to dynamic modeling of a class of mobile manipulators, International Journal of Robotics and Automation, vol. 12, no. 2, pp. 41-48, 1997.

[2] Q. Yu, I. M. Chen, A general approach to the dynamics of nonholonomic mobile manipulator systems, ASME J. Dyn. Syst. Meas. Control 124, 512–521 (2002).

[3] H. G. Tanner and K. J. Kyriakopoulos, Mobile manipulator modeling with Kane's approach, Robotica, vol. 19, pp. 675-690, 2001.

[4] L. Sheng, A. A. Goldenberg, Neural-network control of mobile manipulators, IEEE Transactions on Neural Networks, vol. 12, no. 5, pp. 1121-1133, 2001.

[5] Y. Yamamoto and X. Yun, Unified analysis on mobility and manipulability of mobile manipulators, in Proc. IEEE International Conference on Robotics and Automation, pp. 1200–1206, 1999.

[6] Y. Yamamoto, X. Yun, Coordinating locomotion and manipulation of a mobile manipulator, in Proc. the 31th Conference on Decision and Control, Tucson, AZ, pp. 2643-2648, Dec.1992.

[7] N. E. N. Rodriguez, G. Carbone and M. Ceccarelli, Simulation results for design and operation of CALUMA, a new low-cost humanoid robot, Robotica, vol. 26, pp. 601-618, 2008.

[8] H. I. Christensen, Intelligent home appliances. In Robotics Research, R.A. Jarvis and A. Zelinsky, Eds., no. 6 in Springer Tracts in Advanced Robotics (STAR). Springer Verlag, Heidelberg, Germany, pp. 319-330, January 2003.

[9] M. Lohse, F. Hegel, and B. Wrede, Domestic applications for social robots - an online survey on the influence of appearance and capabilities, Journal of Physical Agents, vol. 2, pp. 21–32, 2008.

[10] T. W. Fong, I. Nourbakhsh, and K. Dautenhahn, A survey of socially interactive robots: concepts, design, and applications, Robotics and Autonomous Systems, vol. 42, no. 3–4, pp. 142–166, 2002.

[11] C. D. Wickens, S. E. Cordon & Y. Liu, An Introduction to Human Factors Engineering, Addison Wesley LONGMAN, 1998.

[12] A. Liegeois, Automatic supervisory control of the configuration and behavior of multibody mechanisms, IEEE Trans. Systems, Man, and Cybernetics, vol. 7, no. 12, pp. 868–871, 1977.