Robotica (1999) volume 17, pp. 529–541. Printed in the United Kingdom © 1999 Cambridge University Press

A manipulation analysis for robot programming

Marco Ceccarelli

Dipartimento di Meccanica, Strutture, Ambiente e Territorio, Universita di Cassino, Via Di Biasio 43, 03043 Cassino (Fr), Italy. E-mail: ceccarelli@ing.unicas.it.

(Received in Final Form: April 21, 1999)

SUMMARY

In this paper a procedure is proposed for a rational and optimum use of robots. It is the result of the lessons learned by the author in the course of his consultancy and teaching activities on manipulative tasks and programming capabilities of robots.

One of the chief aims of this paper is educational, and has been set on the basis of students' practice with robots in the Laboratory of Robotics at the University of Cassino, Italy. A second goal consists in highlighting and teaching how robots' versatility and flexibility can be easily exploited during robot's operation when a task is properly modelled through elementary actions. An elementary action, representing a small manipulative operation and consisting in one or few simple instructions given in robot language, can be easily performed by the robot's programming capability. Some examples have been reported to better illustrate the ideas and the procedure proposed, whose main advantage, also from an educational viewpoint, consist in providing a simpler method for analysing manipulations for robot design and programming.

KEYWORDS: Robot programming; Manipulation analysis; Elementary action.

1. INTRODUCTION

Nowadays robots are widely used, particularly in dangerous and repetitive industrial processes and also for their economic convenience in manufacturing and assembly operations. However, industrial robots are generally considered to be the most complex systems in the field of automation, so that their use is usually restricted to applications requiring users with high level education and continuous training.

This paper reports an attempt to develop a teaching approach for a simple training of robot users based on the fact that the versatility and programming capabilities of a robot can be considered in a complementary light with a view to propose a simple procedure to analyse and program any manipulative task for industrial robots.

This paper's educational aim has been set on the basis on observations made on students' practice with robots in the Laboratory of Robotics at the University of Cassino, Italy; it has also been achieved by using simple and basic assertions with a view to develop a methodology which can be understood and applied by practical engineers, who usually only run robotic systems and are not used to perform analysis of manipulation operations and robot programming.

The key idea proposed in this paper is based on the concept of elementary action, defined as the smallest operation corresponding to the simplest robot action requiring one or few program instructions. By breaking down a manipulative task into a sequence of elementary actions it is possible to deduce flexible programming both with clear manipulative items and program instructions.

Analysis procedures for manipulation such as assembling operations and the related simulations usually result in suitable programming after the sequence of manipulative tasks has been translated into robot language through specific computer packages or by specialised operators. There are some extremely sophisticated computer packages as for example GRASP by BYG Systems Ltd.

In general, the problem of manipulation planning cannot be formulated in an analytical form since rules and constraints, also of a heuristic nature, are often impossible to express by simple formulas. Several specific and advanced techniques are available both in the literature and commercial market for off-line robot programming: for example, assembly sequence has been planned by using CAD-CAM systems to evaluate and simulate the capabilities and behaviour of robots during operation,[1,2] or to model and design an object-oriented robot programming;[3] by using suitable graphs which can be optimised by means of genetic or fuzzy logic algorithms;[4,5] or by using algorithms which are based on Petri nets, like in reference 6.[6]

In addition, for practical reasons, great attention has been paid to a simple and effective way of teaching robot programming as is also indicated in reference 7 in which a Computer Assisted Learning method for robot programming has been proposed with its attribute of simplicity.

The educational theme of this paper aims at answering the need that has emerged in the teaching of robotics and robot programming basics for simple and clearly understandable methods for manipulation analysis. The simple methodology proposed has been and can be useful also in industrial applications because it enables industrial managers to understand how to improve robotized workcells productivity and/or efficiency by using procedures with easy physical interpretations.

The author's experience in the field of teaching and consulting has been reported in this paper with a view to better explain the proposed procedure and the advantages of using it. Particularly, comments have been reported to

https://doi.org/10.1017/S0263574799001745 Published online by Cambridge University Press

highlight the procedure's simplicity and practical feasibility, also supported by a numerical evaluation of actual efficiency.

MANIPULATION AND ROBOT REQUIREMENTS

A robot has been defined by ISO as a versatile and re-programmable device capable of performing a variety of manipulative tasks. Some other definitions with small conceptual differences have been proposed as is the case with IFToMM, defining a robot as a,[8] "mechanical system under automation control that performs operations such as handling and locomotion". On the other hand, manipulation can be thought of as "controlled movement of object".[8]

However, it is commonly accepted that the basic properties for an automatic machine to be considered a robot are mechanical versatility and programming flexibility, as for example in references 9, 10. These features must both belong to a robot, otherwise a device with mechanical versatility only is a manipulator, and a machine with programming flexibility only is an automatic device. Thus, for example, a CNC lathe may be very sophisticated electronically and be a very flexible machining device, however it can never be considered a robot since it has no mechanical versatility to perform operations other than turning. By the same token, a mechanical manipulator capable of performing several kinds of movements cannot be considered a robot unless it is equipped with the electric and electronic devices needed for programming capabilities. Indeed, mechanical versatility consists in a machine's capability to perform different manipulative tasks thanks to the properties of its mechanical structure and in particular because of the kinematic chain. Therefore, versatility depends on mechanical design and structure and it can be modified only with structural or dimensional changes. In particular from a technological viewpoint, versatility is strongly related to the Mechanics and Mechanical Engineering of the manipulator structure.

On the other hand, programming flexibility consists in the capability to perform different manipulative tasks thanks to the properties of the programming language used, and is particularly the result of instructions and sequence updating. Hence, from a user viewpoint flexibility depends on software capability, whereas from a designer viewpoint it depends on hardware capacity.

Following the aforementioned considerations, versatility and flexibility in a robot are complementary in the sense that the main advantages of one aspect can be fully exploited only when the other has been properly designed and used.

Therefore, for a rational and optimum use of a robot the basic problem lies in how to exploit a robot's two fundamental features in a complementary and simultaneous way.

A PROCEDURE FOR MANIPULATION ANALYSIS

A key issue for an optimum use of robots lies in the fact that a robot should be introduced to a user by stressing that versatility and flexibility are equally important: too much attention on one of the two aspects only may give the wrong impression and lead a user to think that one aspect predominates and that it may solve any deficiency due to the other one.

Therefore, from an educational perspective, the best approach consists in imparting knowledge separately on the manipulative process to be performed and on the robot to be used.

Thus, it was found convenient to study manipulative tasks without any reference to any particular robot in order to determine manipulation requirements independently from the limits of a specific robot. By the same token, a robot's capabilities can be conveniently learned in terms of programming facilities and mechanical capabilities with no reference to specific applications, which may reduce the interest for overall capabilities.

At this stage the problem of matching the results of learning activities and analysis arises, consisting in finding a feasible way to make the robot perform a given manipulative task with the required flexibility and versatility. Thus, this way a user can gain an insight into how to use rationally a given robot for a given manipulation, and particularly into how to further improve with minor adjustments the robot's operation or industrial process in terms of productivity, efficiency or changes in manufacturing production.

In Fig. 1 the aforementioned considerations have been translated into a practical procedure used by the author for students' practice with robots within a university course at the Laboratory of Robotics in Cassino. Fig. 1 indicates the steps required to analyse a manipulative task in order to identify a specific robot and operate it through a suitable programming which can be verified for flexibility and versatility optimum performances. In particular, according to item one of the flowchart in Fig. 1 a given process has to be analysed in terms of manipulative operations, even mimicking human actions in order to establish rigorously what manipulations are required.

Then, the environment should be studied or designed to identify obstacles and constraints on manipulation. This may yield a geometric and kinematic model of the environment the robot will work in. At this stage students-users are encouraged to think in a more abstract way in the sense that the process should be studied with no reference to a predetermined automated or robotized solution.

According to the next item in the flowchart, process operations have to be identified as proper manipulative actions, i.e. movements of objects including tools and subsidiary machines. This can also be done by drawing sketches for the robotized lay-out to identify manipulative tasks. This modelling activity, even through drawing sketches, will help check both manipulation and robot performances.

The next two steps deal with a more detailed manipulation analysis with the aim to identify elementary actions and qualify their nature and timing. In the next section this analysis is discussed in greater detail since it constitutes the core of the procedure and the main novel contribution of this paper.

Once the manipulation entities have been identified and portioned, it has to be established whether the robots capabilities are suitable for the operation's purpose. In this

https://doi.org/10.1017/S0263574799001745 Published online by Cambridge University Press

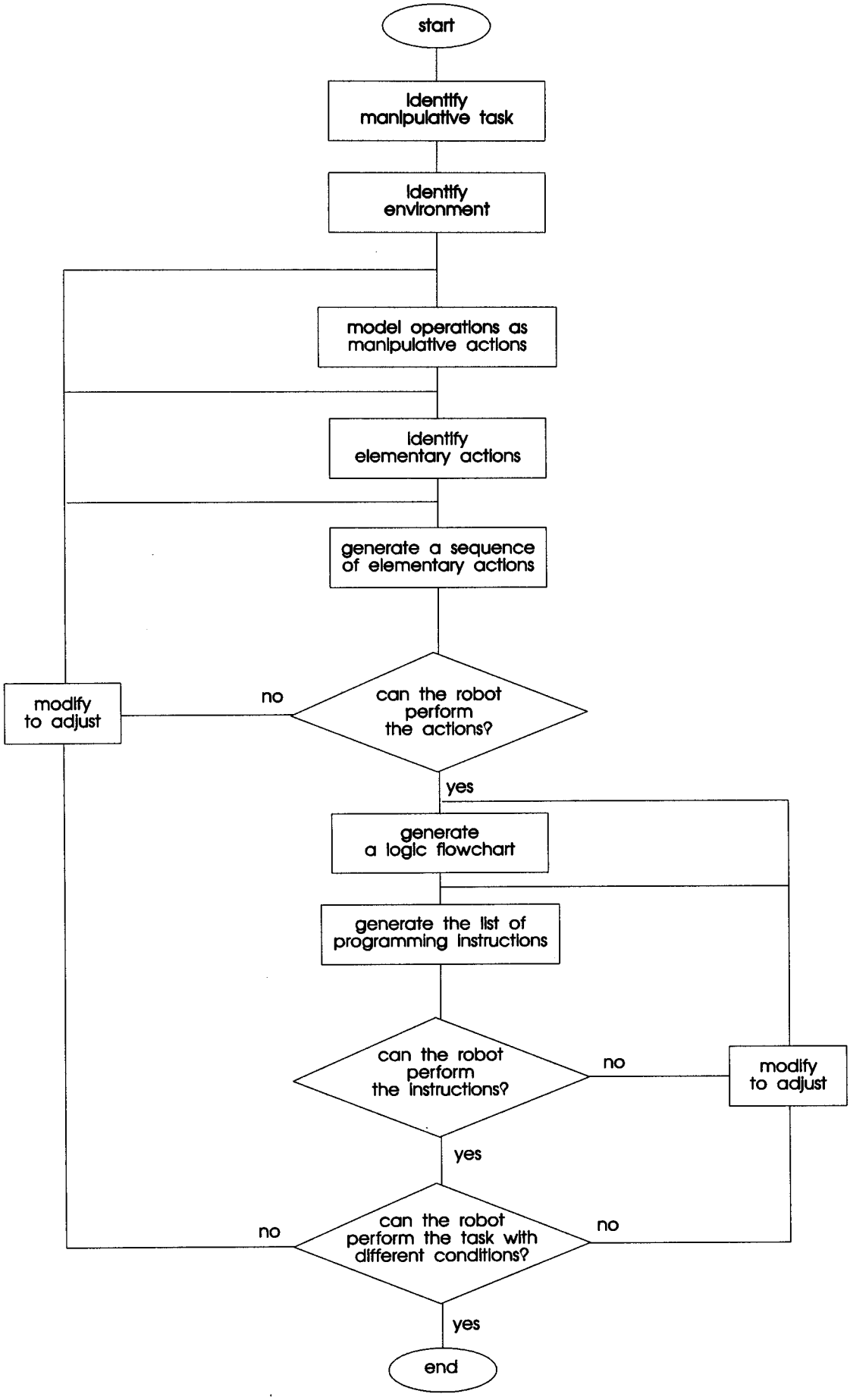
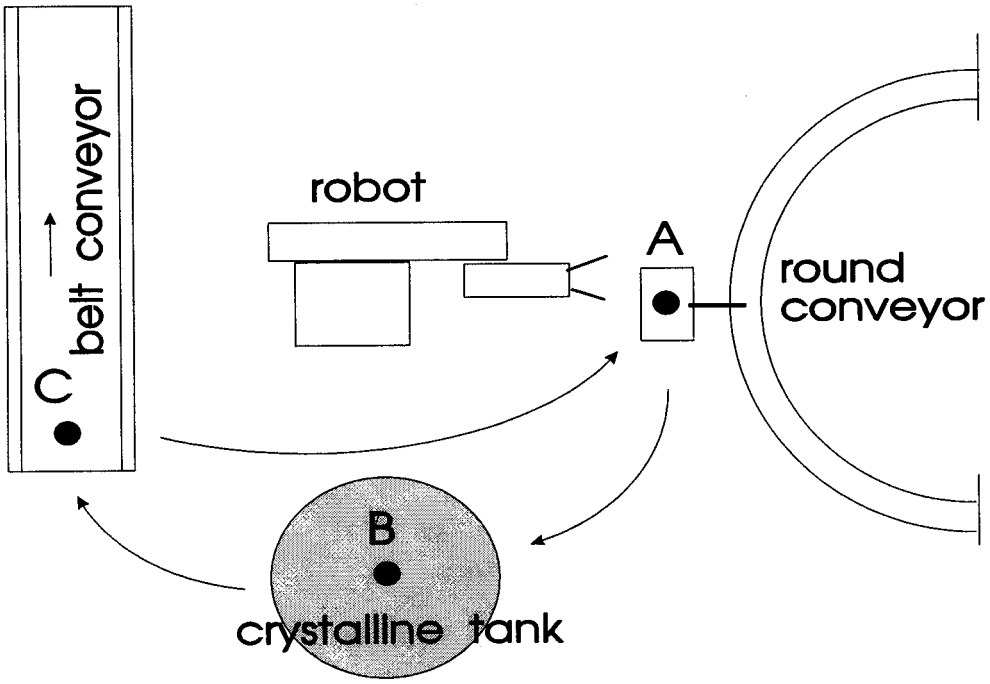


Fig. 1. A procedure for manipulation analysis and robot programming.

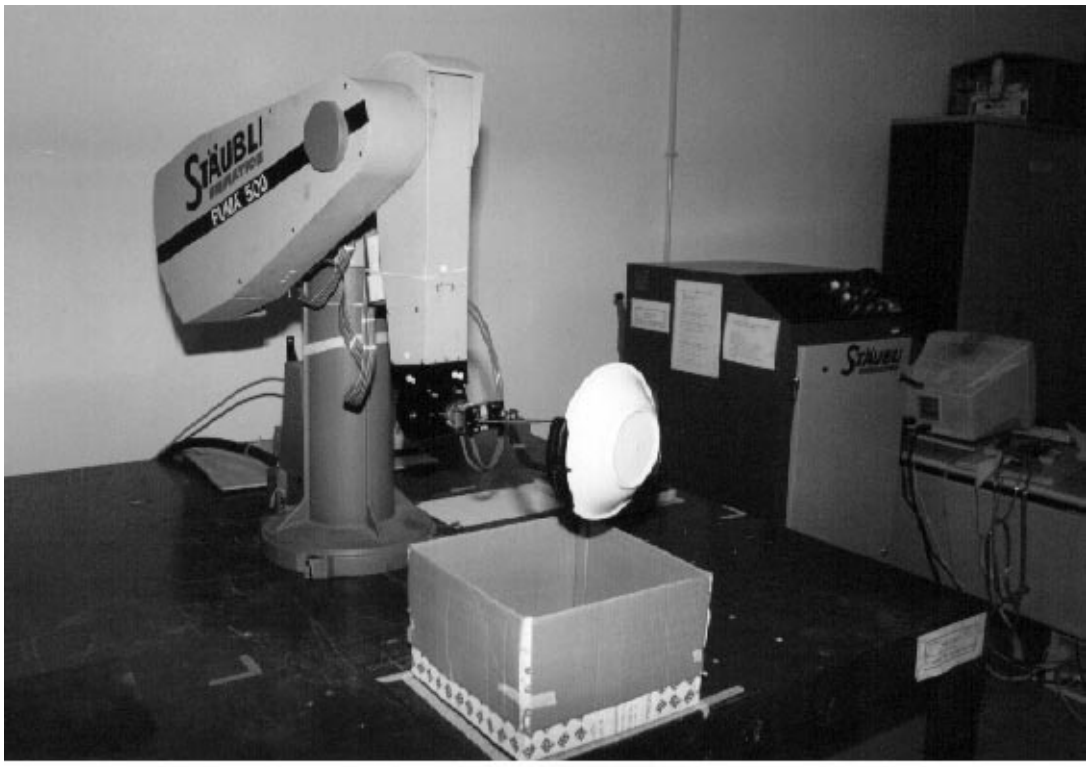
https://doi.org/10.1017/S0263574799001745 Published online by Cambridge University Press

item robot performances are verified with respect to the given manipulation. This step can be useful either to check operation feasibility of a given robot or to select a better robotized solution. In Fig. 1 a first adjustment of previous steps is required following an unsatisfactory response, i.e. when the given robot or the feasible robot cannot comply properly with the previously obtained manipulation sequence. Adjustments may consist in rethinking the modelling of the manipulative tasks by changing also elementary actions and/or the design of the elementary actions sequence.

The item "modify to adjust" can be developed at different levels of detail and it may describe and include specific algorithms to obtain optimum solutions for each specific



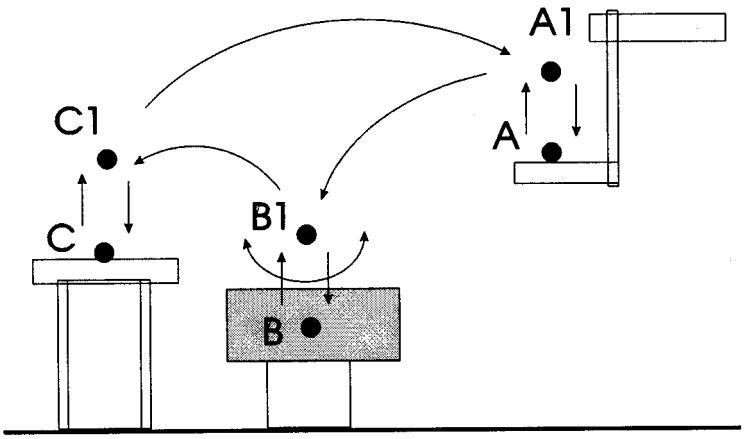
(a)



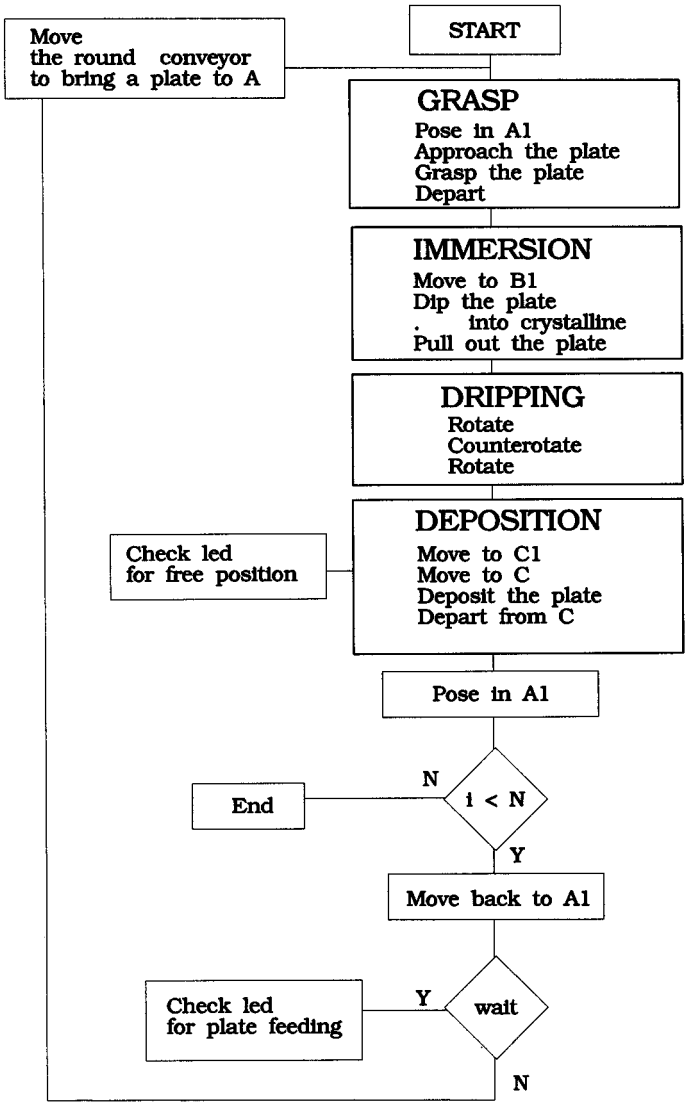
(b)

Fig. 2. An illustrative example of the application of the proposed manipulation analysis in students' practice with PUMA562 robot[12]: (a) an analysis of the environment; (b) the practising layout.

https://doi.org/10.1017/S0263574799001745 Published online by Cambridge University Press



(a)



(b)

Fig. 3. Continued.

https://doi.org/10.1017/S0263574799001745 Published online by Cambridge University Press

Elementary actions robot	T	M	A	P	time [sec.]
goes to A		•			0.2
grasps a plate			•		0.2—0.5
goes to B	•				2.5
dips the plate	•			•	1.0
pulls out the plate	•				0.5
moves to drip	•		•		2.2–2.5
goes to C	•				2.5
deposits the plate	•		•		1.0
goes back to A		•			1.0
checks for the plate			•	•	0.5–2.5

(c)

Fig. 3. Results from a manipulation analysis and robot programming for the case of Fig.2: (a) a model of the manipulative trajectory; (b) logic flowchart of elementary actions; (c) an evaluation of elementary actions.

problem. A manipulation analysis is a complex task and it includes several difficult problems that can be solved separately with suitable techniques within a common frame, so that constraints on one aspect can be the result of another one and vice versa. In this sense each main block in the Fig. 1 procedure is completed by the item “modify to adjust” which is connected to. In addition this item specifies how to obtain the modifications which are usually evaluated by means of sophisticated algorithms for any specific problem. In this paper we do not make use of these algorithms since we want to focus on manipulation analysis for teaching purposes. Therefore, at a first level of analysis this approach can be conveniently thought as of an empirical nature. However, the item “modify to adjust” encompasses a series of elements, such as considerations, computations and evaluations, which are useful to provide better solutions for a problem that has not been considered or solved in a feasible or acceptable way.

However, it is worth pointing out that to solve difficult problems suitable techniques and procedures are available in the literature, such as robotized workcell lay-out design, choice and optimal placement of an appropriate robot, optimal planning for robot trajectory and object assembling. Therefore, the proposed manipulation analysis can also be considered as a common frame capable of grouping these different problems and consequently connect their partial solutions and constraints within a unique problem solving approach.

Once the robot is considered feasible because its versatility matches the manipulation requirements, a programming design for a flexible solution is started by developing a logic flowchart for programming instructions. In my experience, the flowchart has proved to be very useful in teaching since students have had a better response (quicker and easier) in adjusting the programming when later changes in the manipulative process are required. This is true, indeed, for any software design: in a software program a flowchart may be of great help for future updating of instructions, and also helps susequent users understand. Furthermore, this way a user obtains a programming solution which is not yet directed to a given robot, because it is not oriented to a given programming language. Indeed, the logic flowchart can be used to program manipulations in several different robot languages. This obviously means that the flowchart is independent of the robot choice.

The next step consists in generating a list of programming instructions for a given robot with the instructions of a given programming language, which requires knowledge of instructions syntax as well as the limits of robot movements and performance.

According to the item “can the robot perform the instructions?” a first check on the feasibility of the programming instructions sequence may require adjusting the previous programming solutions.

Finally, the flexibility of the robotized solution can be verified by introducing some changes in the manipulative process. This will show whether the robot’s versatility allows it to operate properly with some further change. Although this last item may not be necessary for a given application, it is however of primary importance for an optimum use of robots because it reveals how versatile and flexible a designed robotized solution is.

The above mentioned checks can be performed by a debugging process as well as by means of on-line practical tests. Furthermore, for an optimum (versatile and flexible) use of a robot, the validation items can also suggest adjustments in the robot structure or particularly in the robot end-effector, so that gripper design[11] can be included in the analysis of the manipulation.

A MANIPULATION ANALYSIS FOR ROBOT PROGRAMMING

A manipulation is a process performing “controlled movements of object”.[8] To achieve maximum efficiency in robot operation this process, which can be very complex, has to be analysed in depth by breaking down the manipulation into manipulative elementary tasks – defined as the smallest actions, and named elementary actions – to be performed for object and robot movements. The object can be a grasped object or the robot end-effector itself since the manipulative goal may require intermediate and subsidiary actions on the robot. Basically, in the first stage of a manipulation analysis only the movements of the robot end-effector or of the object have to be taken into account, with no reference to robot capabilities nor to human mimic,

https://doi.org/10.1017/S0263574799001745 Published online by Cambridge University Press

although this last comparison may help to understand the complexity or simplicity of the manipulative task. In practice this can be achieved by looking at the movements of a rigid body representing either a grasped object or the robot end-effector.

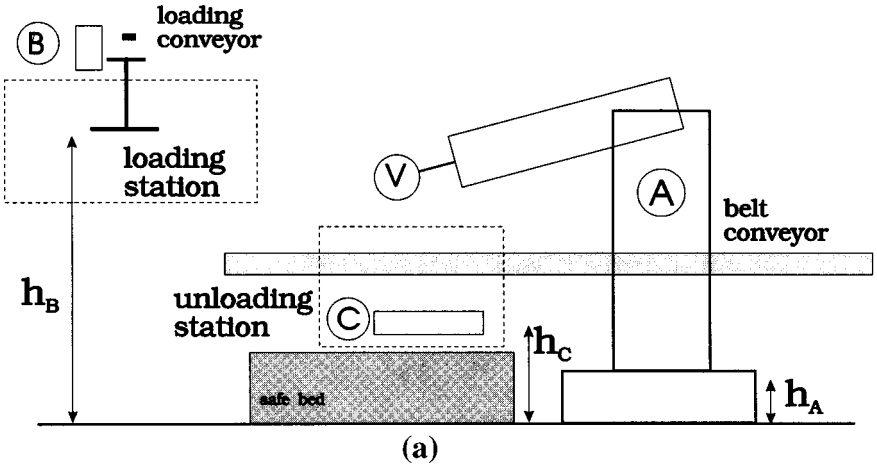
The identification of the elementary actions performed for the rigid body of the reference object to be manipulated, provides a useful tool to break down the manipulative process.

A first result may consist in an easy evaluation of the number of degrees of freedom required to perform the process. This constitutes a basic criterion for choosing the appropriate robot or a class of robots for a given manipulation.

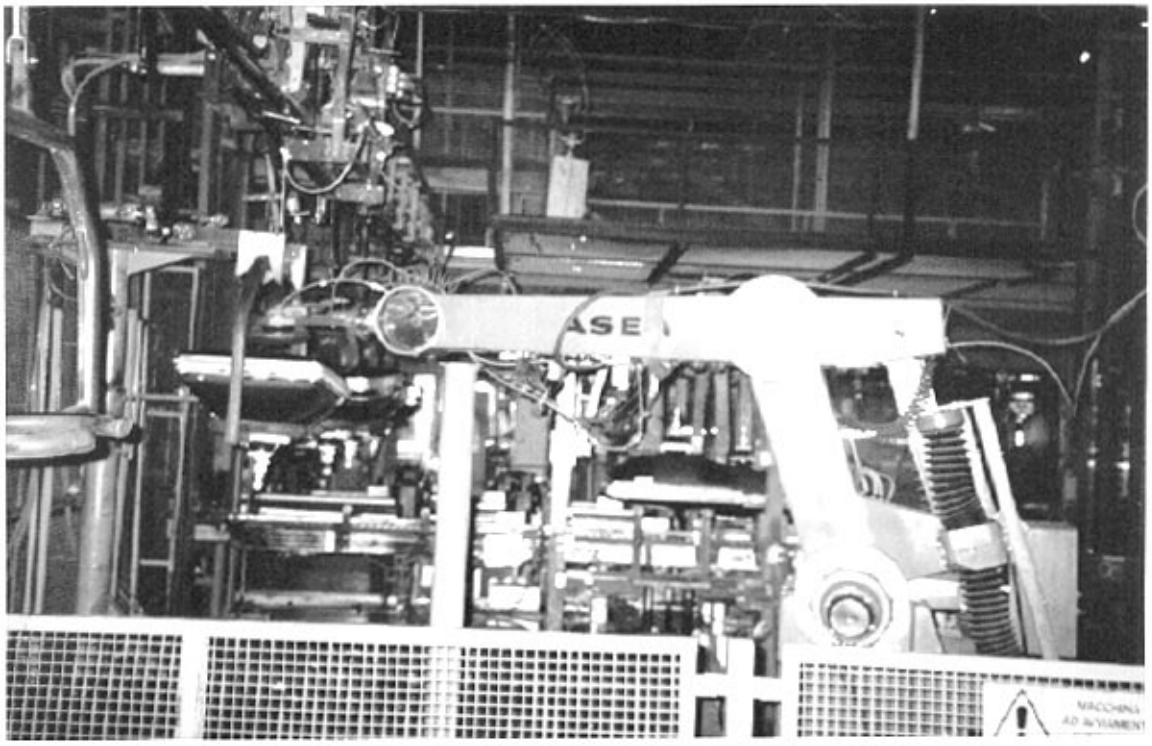
At this stage a schematic drawing of the object's fundamental positions and configurations is particularly useful. This allows the working area and the mechanical versatility of a robot to be easily determined thus helping in rationally choosing the appropriate robot.

Once the process has been analysed in terms of elementary actions, the robotized portion can be selected and a further analysis may be of help in carrying out flexible programming as Fig. 1 illustrates.

The elementary actions to be performed by a robot can be conveniently identified and classified as transport, movement, passive pause, active pause. A transport elementary action T can be defined as a robot motion performing a movement of the object which is transported from one position to another. A movement elementary action M can be defined as a robot motion performing a change of the



(a)



(b)

Fig. 4. An illustrative example of the application of the proposed manipulation analysis in an industrial kinescope manufacturing[14]: (a) an analysis model of the robotized workcell; (b) ASEA IRB60 robot manipulating a kinescope. (Courtesy of Videocolor Ltd).

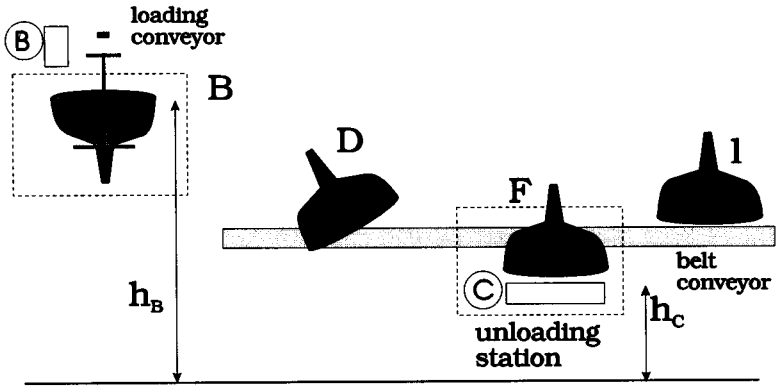
https://doi.org/10.1017/S0263574799001745 Published online by Cambridge University Press

manipulator configuration to obtain a movement of the robot end-effector, when this is not considered the primary object to be manipulated in the process. A pause can be considered an elementary action of the robotic system during which no movements are performed but some other activities are undergoing. A passive pause P can be considered when no activity is performed and the robot is waiting to start again. The robot is in a state of active pause A when it is not moving but is operating through its equipment (sensors, control, programming unit, power supply).

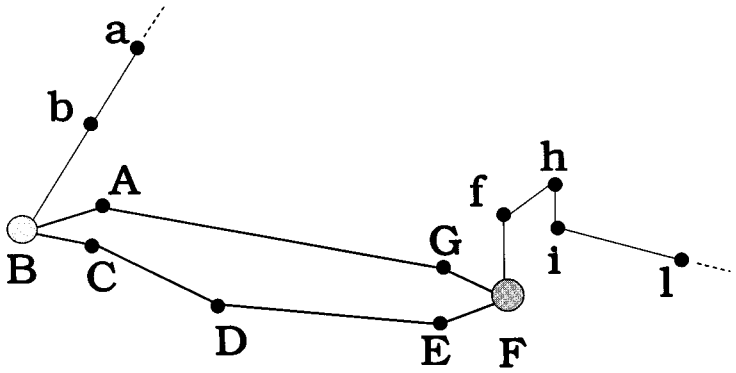
Once all elementary actions have been identified and classified, they can be organised or re-organised for a suitable manipulation sequence also by grouping them in some set of actions, which may perform a well identified sub task. This can be useful if parts of this sub task are repeated within the process.

Subsequently, a user can generate the programming list by considering that each elementary action may correspond to one or few instructions. The logic flowchart can be of great help for future adjustment/improvement of the robot operation also with respect to changes in the programmed manipulation, since it allows to easily identify the instructions or the set of instructions to be modified or changed.

For an optimum use of a robot a numerical evaluation of the performed manipulation can be obtained by recording



(a)



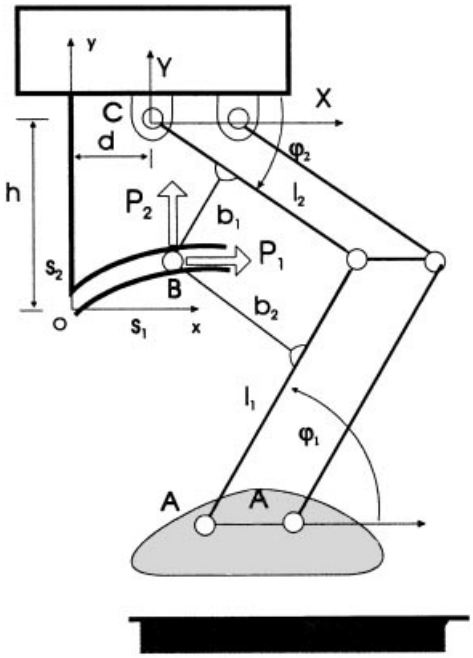
(b)

Elementary actions robot	ref. points	T	M	A	P	time [sec.]
waits to start	A—A				•	0—0.5
approaches loading conveyor	A—B		•			1.0
activates suction cup end-effector	B—B		•	•		0.1—0.5
departs from loading conveyor	B—C	•				1.0
departs from loading station	C—D	•				1.0
goes through waiting point	D—D	•			•	0—0.5
approaches the unloading station	D—E	•				1.2
approaches the unloading plane	E—F	•				1.3
deposits the kinescope	F—F			•		0.2
dearts from the unloading plane	F–G		•			0.7
goes to the starting point	G—A		•			2.0

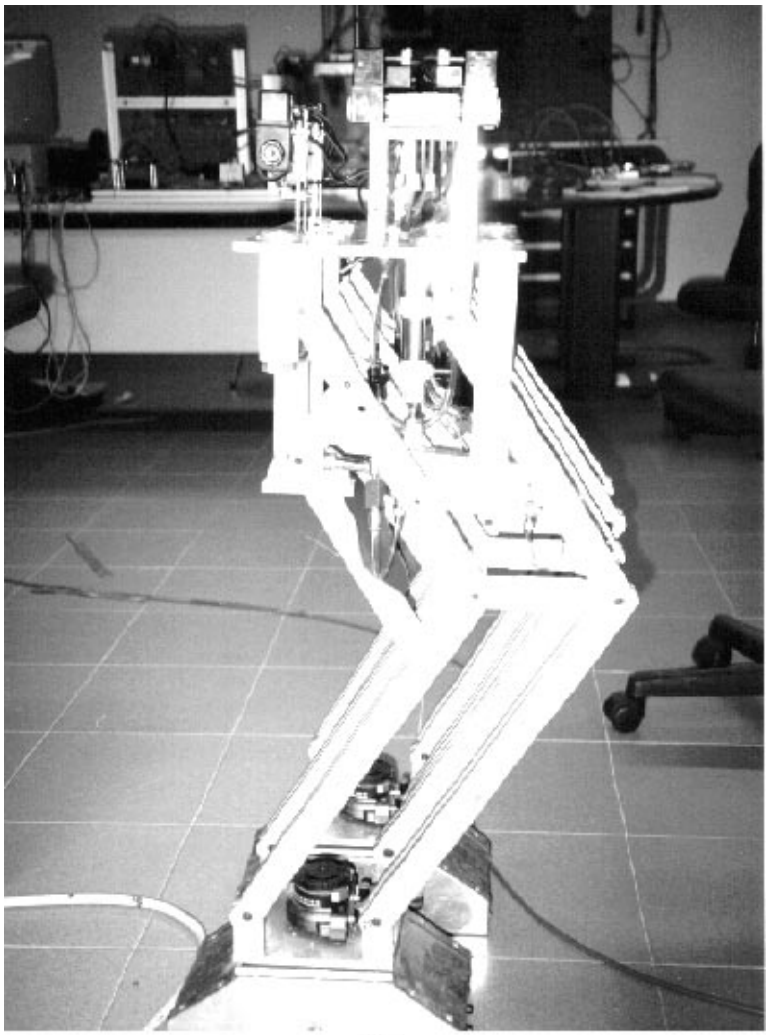
(c)

Fig. 5. Results from a manipulation analysis and robot programme for the case of Fig. 4: (a) a model for the present manipulation of a kinescope; (b) a model of the manipulative trajectory; (c) an evaluation of elementary actions of the robot operation.

https://doi.org/10.1017/S0263574799001745 Published online by Cambridge University Press



(a)



(b)

Fig. 6. An illustrative example of the application of the proposed manipulation analysis in the operation of a biped walking robot,[15]: (a) a kinematic model for the leg mechanism; (b) the prototype EP-WAR built at the Laboratory of Robotics in Cassino[16].

https://doi.org/10.1017/S0263574799001745 Published online by Cambridge University Press

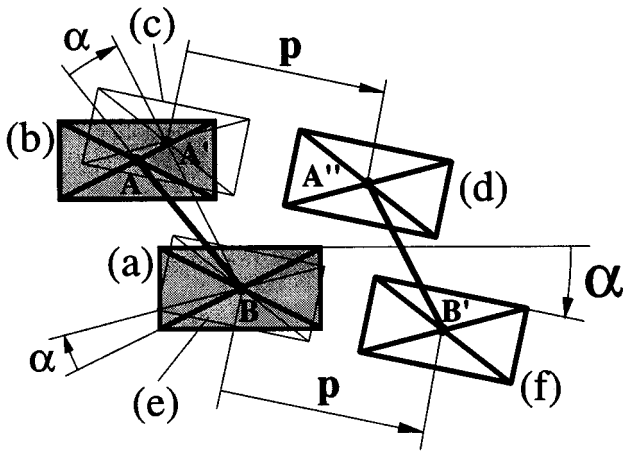
the operation time for each elementary action. This gives an assesment of the robot operation and a tool to identify possible optimisation with respect to productivity and manipulation changes.

Indeed, several other criteria have been proposed in the literature and are also currently used to optimise, for example, energy consumption, actuator torques, velocity and acceleration values, and operation cost. Only cycle time has been used in this paper since it is of evident interpretation and interests for a practical synthetic evaluation.

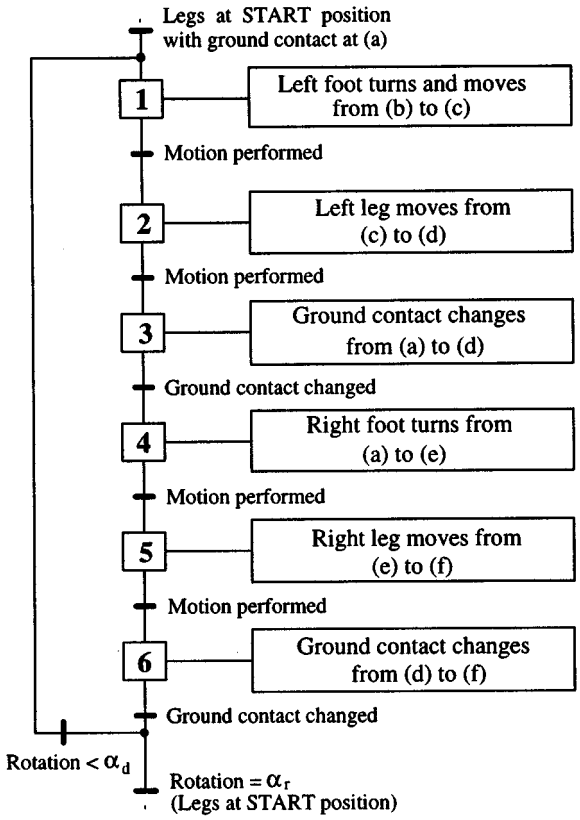
Some examples have been reported in the next section to illustrate in detail the application of the proposed ideas and the feasibility of the proposed manipulation analysis.

SOME EXPERIENCES AND ILLUSTRATIVE EXAMPLES

The proposed procedure for manipulation analysis and robot programming has been used by students with a PUMA562 and a Scorbot ER-Vplus robot at the Laboratory of Robotics



(a)



(b)

Fig. 7. Continued.

https://doi.org/10.1017/S0263574799001745 Published online by Cambridge University Press

Elementary actions left leg	M	A	P	time [sec.]	M	A	P	elementary actions right leg
waits			•	0.1		•		starts suction cups
turns foot	•			0.2		•		starts suction cups
moves to A'	•			0.2		•		starts suction cups
moves to A''	•			0.5		•		starts suction cups
starts suction cups		•		0.1		•		starts suction cups
starts suction cups		•		0.1		•	•	stops suction cups
starts suction cups		•		0.2	•			turns the foot
starts suction cups		•		0.5	•			moves the foot to B'
starts suction cups		•		0.1		•		starts suction cups
stops suction cups		•	•	0.1		•	•	waits

(c)

Fig. 7. Results from a manipulation analysis and robot programming for the case of Fig.6: (a) a model for the manipulative turning trajectory of biped walking; (b) a logic flowchart for a modular programming; (c) an evaluation of elementary actions performing a turn while walking.

at the University of Cassino since 1994. Figures 2 and 3 illustrate a students' practice simulating an industrial process concerning the immersion of a ceramic plate into a recipient with crystalline fluid as a final step of a plate production.[12]

Particularly, following the procedure of Fig. 1, once the manipulation task has been identified by referring to a current manual operation, a robotized workcell has been designed. The environment has been specified as the robotized workcell of Fig. 2 (a), where a robot has been located in a central position with respect to a round feeder, the crystalline tank and an unloading belt conveyor.

After the manipulation task has been identified with respect to manual operation (Fig. 1), a robotized workcell has been designed and its environment specified as is the case with the robotized workcell of Fig. 2 (a), where a robot has been placed in a central position with respect to a round feeder, the fluid tank and an unloading belt conveyor.

The criterion for placing the robot in this position results from the manipulation analysis carried out, aiming at optimising robot size and manipulative trajectory through empirical adjustments during laboratory experiments. In this students' practice optimal trajectory planning has not been set on the basis of a specific analytical formulation but it has been estimated by only taking into account the robot movements in a general way. The practising layout at the Laboratory of Robotics in Cassino is shown in Fig. 2 (b), in which a PUMA robot has been used because of its six d.o.f. (degree of freedom) which have been computed as needed to perform the prescribed actions. The manipulation analysis has been carried out according to the procedure of Fig. 1 and the results have been reported in Fig. 3. Fig. 3a) shows a model for the manipulative actions in which the reference points A, B, C referring to those of Fig. 2a) have been indicated with the manipulative trajectory between them. In particular, points A_1, B_1, and C_1 have been specified as those reference points from which accurate motion of the robot starts to final positions A, B, and C respectively.

Manipulations have been detailed in the form of elementary actions and the table of Fig. 3b shows a sequence of the operations. Particularly the dripping motion at point B1 in Fig. 3a has been prescribed in the form illustrated in Fig. 3b after some experiences made following the item "modify to adjust" of the procedure of Fig. 1 in order to ensure appropriate draining and uniform distribution of the crystalline liquid on the plate's surface. Number N refers to the number of plates in a production cycle and it can be prescribed for continuous functioning. Two led checks have been designed in the workcell to have the robot pause in positions suitable both for the manufacturing process and the robot. A first led gives information on free place in the unloading conveyor and it controls a passive pause of the robot. The second led sensor recognises the presence of a place in the round feeder and gives the start signal to the robot.

Fig. 3b lists elementary actions by grouping them in appropriate operations which are a function of one reference point each so that programming flexibility depends on the reference points co-ordinates only.

Finally, Fig. 3c reports a final evaluation of the experienced practices carried out also in terms of operation timings in which some elementary actions have been grouped according to Fig. 3b. The nature of the elementary actions have been indicated in table of Fig. 3c according to the classification of the previous section as movement M, transport T, active pause A and passive pause P. This has been used to optimise both the manipulation and the programming not only for cycle time. The time evaluation can be very interesting since it gives a measurement of the length of each action within the overall cycle; moreover, it can be used to identify what operation can be conveniently modified to improve productivity. Thus, from Fig. 3c it is evident that the dripping operation can be easily speeded up, albeit only within a given range ensuring proper draining. The transport operations to B and C can also be speeded up but in a different way since the last transportation to C concerns a plate covered with crystalline and excessive acceleration may alter the crystalline's distribution on the plate.

The same procedure for manipulation analysis has been used by the author and his co-workers for industrial applications under research contracts.[13] In particular, in this paper we refer to a study of feasibility for productivity improvements and reduction of robot failures in a kinescope

https://doi.org/10.1017/S0263574799001745 Published online by Cambridge University Press

manipulation at the Videocolor plant in Anagni.[14]

The robotized workcell is shown in Fig. 4, where automated equipment includes a suitable conveyor for the loading and a specific station for unloading. The robot is positioned to pick up a kinescope from the conveyor and carry it to the unloading station after rotating it 180 degrees. In this case, following the procedure of Fig. 1 the operation of the robot and overall workcell has been studied to determine possible source of failures and optimise cycle time by identifying which operations can be easily speeded up. Thus, the required manipulation of a kinescope from a loading position to the unloading station has been first analysed with no reference to robotization by focusing on object-kinescope locations, Fig. 5a, and the trajectory of the robot end-effector, Fig. 5b. Therefore, the functioning of the workcell has been analysed and the characteristic points A to E have been determined in the manipulative trajectory of a kinescope performed by the robot, Fig. 5a. Similarly, characteristic points a, b have been identified for the operation of the loading conveyor and points f to l for the operation of the unloading station and unloading belt. These points have been considered and used as reference points both for manipulation analysis and robot programming. The present robot programming has been revised by referring to the manipulative trajectory of Fig. 5b so that the elementary actions of the robot have been identified. The functioning of the workcell and the robot has been evaluated several times during industrial production, also by video recordings, and an average timing has been estimated for a regular working as the one reported in Fig. 5c. The main results of this manipulation analysis for an industrial application consist in the fact that the robot's better positioning in the workcell and with respect to the loading and unloading locations will improve both the efficiency of the robot and the productivity of the workcell. In addition, from a manipulation viewpoint it has been estimated that the use of the reference point D as a waiting point for the robot manipulation can be a serious source of robot failures because it requires the robot to start with a critical motion owing to both the rotated position of a kinescope and the robot configuration. The end-effector trajectory has been analysed by taking into account the movements of the robot as function of the manipulation content and cycle time in the list of Fig. 5c, so that their importance has been qualified and easily recognised with the aim to improve productivity and decrease the critical aspects of robot configurations.

The manipulation analysis procedure of Fig. 1 has also been used in research design and in setting up a new biped walking robot. In particular, the proposed procedure has been applied to design electropneumatic actuation of the walking robot and to develop a modular programming of the walking motion in a new biped robot built at the Laboratory of Robotics in Cassino.[15]

The manipulative task and the environment have been prescribed as human-like walking on a smooth surface (as is the case in industrial plants) to obtain the mechanical design of a prototype named as EP-WAR,[16] illustrated in Fig. 6 by the kinematic structure of a leg mechanism in Fig. 6a and the built prototype in Fig. 6b. Figure 6a also reports a kinematic analysis of the working of a leg during walking. Thus, actuator actions P_1 and P_2 will move the pantograph and the double parallelogram of a leg so that the foot will be lifted and moved straight forward. Indeed, the motion of point B is scaled with ratio 1:4 to motion of point A because of the pantograph linkage. To perform forward motion of the body, and make the biped robot walk, the two legs actuators have to be co-ordinated: one leg – grasping the terrain because of underfoot suction cups – pushes forward the body and at the same time the other leg goes forward according the scheme of Fig. 6a. This last movement is modelled in Fig. 6a by considering the manipulation actions and a similar sketch can be drawn for the pushing forward the body.[15,16]

In particular, manipulation analysis has been applied to develop suitable programming for the action of walking in the form of modules appropriate for straight and circular paths, so that an easy combination of them can obtain any path in the walking of the robot.[12] First, the manipulative trajectory has been modelled as in Fig. 7a where the robot's feet are depicted while performing a walking step sequence. The sequence has been ordered from position (a) to (f) in the figure. Analysis has yielded the logic flowchart of Fig. 7b which has been successfully used for the development of a Grafcet list of programming instructions of the PLC which controlls the actuation. In this flowchart it is easy to recognise elementary actions referring to the feet positions of Fig. 7a. The experimental validation of both prototype running and manipulation analysis procedure has been successfully carried out by measuring walking performance time, as shown in Fig. 7c. Currently, the walking movement is slow, as stressed by the values of Fig. 7c; however, it can be speeded up following the last items of the Fig. 1 procedure.

Figures 2 to 7 show the soundness and the feasibility of the proposed manipulation analysis which has been successfully utilised in applications in the fields of teaching, industrial manufacturing and research prototyping.

CONCLUSION

In order to achieve optimum operation in robots, the robotic systems' versatility and flexibility have to be both exploited. This may require an analysis of the manipulative process to be carried out with no reference to the robot to be used. These two aspects can be subsequently put together by using the manipulative capability of a robot in terms of mechanical versatility and programming flexibility through a manipulation analysis in terms of elementary actions. This paper presents an original procedure for manipulation analysis based on the concept of elementary action.

The procedure has the advantage of being simple, and therefore suitable for teaching, and yielding results comparable to those which can be obtained with more sophisticated procedures.

Some examples have been reported to better illustrate the ideas and the feasibility of the proposed procedure by stressing its simplicity and usefulness as a teaching tool.

ACKNOWLEDGMENTS

The author wishes to thank Dr Giorgio Figliolini of the University of Cassino for his collaboration in developing the

https://doi.org/10.1017/S0263574799001745 Published online by Cambridge University Press

project of a biped walking robot and the students of the course on Mechanics of Robot in the years 1994-1997 for the enthusiasm displayed during their practice with robots at the Laboratory of Robotics in Cassino.

References

1. E. Ceresole, "CAD System for Sequence Planning of Robotized Assembly", Ph.D Thesis (University of Brescia, 1996). (In Italian).
2. A. Liegeois, Borrel, E. Dombre, "Programming, Simulating and Evaluating Robot Actions", in: *Robotics Research* (The MIT Press, Cambridge, 1984), pp. 411–418.
3. C.H. Fedrowitz, "An Object-Oriented Approach for Improving Design and Portability of Robot Offline Programming and Simulation Systems", *Proceedings of the 27th International Symposium on Industrial Robots*, Milan (1996), pp.781–786.
4. M.F. Sebaaly, H. Fujimoto, "General Assembly Planning Automation with Fuzzy Graph Representation and Genetic Search", *Proceedings of the World Automation Congress WAC'96*, Montpellier (1996), **Vol. 5** – Soft Computing with Industrial Applications, pp. 589–596.
5. S. Shu, B. Zhang, D. Liu , J. Xu, "Research on Robotic Assembly Process Analysis and Evaluation", *Proceedings of the 3rd Asian Conference on Robotics and Its Application* ACRA (Tokyo), 1997, pp. 115–120.
6. M. Jinyan, S. Y. Chai, W. Youyi, "On-Line Scheduling Using Coloured Petri Net for FMS", *Proceedings of the 26th International Symposium on Industrial Robots*, Singapore (1995), pp. 437–442.
7. R. Maglica, N. Martensson, "Teaching Robot Programming Using CAL", *Proceedings of the 26th International Symposium on Industrial Robots*, Singapore (1995), pp. 111–116.
8. IFToMM Commission A, "Standard for Terminology", *Mechanism and Machine Theory*, **26**, No. 5, 526 (1991).
9. R.D. Klafter, T.A. Chmielenski, M. Negin, *Robot Engineering: An Integrated Approach*, (Prentice-Hall, NewYork, 1989).
10. L. Sciavicco, B. Siciliano, *Modelling and Control of Robot Manipulators*, (Mc Graw-Hill, New York, 1995).
11. M. Ceccarelli, "Design Problems for Industrial Robot Two-Finger Grippers", *3rd International Workshop on Robotics* in Alpe-Adria Region, Bled (1994), pp.117–120.
12. G. Cortina G., "Plate immersion with PUMA562 robot", *Practice Report, Laboratory of Robotics* (Cassino, 1995) (in Italian).
13. M. Ceccarelli, F. Iacobone, L. Carrino, E. Anamateros, "On the Manipulation of a Composite Material Roving for a 3D Winding Manufacturing", *26th International Symposium on Industrial Robots*, Singapore (1995), pp. 597–602.
14. M. Ceccarelli, "Manipulation Analysis and Optimization for Using Robots in an Industrial Manufacturing", *Internal Report under Research Contract with Videocolor SpA, Laboratory of Robotics* (Cassino, 1997) (in Italian).
15. G. Figliolini, M. Ceccarelli, "Easy Programming of an Electropneumatic Walking Robot", *5th International Symposium on Robot Control SYROCO'97*, Nantes (1997), **Vol. 3**, pp. 793–800.
16. G. Figliolini, M. Ceccarelli, "A Mechanical Design of an Electropneumatic Antropomorphic Walking Robot", *11th CISM-IFToMM Symposium on Theory and Practice of Robots and Manipulators Ro.Man.Sy.'96*, Wien (1996), pp. 189–196.

https://doi.org/10.1017/S0263574799001745 Published online by Cambridge University Press