INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07
28 - 31 AUGUST 2007, CITE DES SCIENCES ET DE L'INDUSTRIE, PARIS, FRANCE

ENABLING DESIGN CONSTRAINTS IN COMMERCIAL CAD SYSTEMS

B. Singh[1], J. Matthews[1], G. Mullineux[1] and A. J. Medland[1]
[1]University of Bath, UK

ABSTRACT

Computer aided design (CAD) systems have been helpful in automating many of the activities involved in a typical design process. The advent of constraint based CAD systems has made these systems more intelligent. However the use of constraints in a typical CAD system is often restricted to the identification and resolution of geometric constraints. Another limitation of these systems is that they cannot be customised for specific needs as they are closed and its difficult to know what is going on inside. This means that the designer cannot add further constraints into the system. Constraint based techniques on the other hand provide such flexibility of customisation. These techniques have been proved useful in mechanism and machinery design. Using them in conjunction with the features offered by a typical CAD system can provide support for tackling higher and more general form of constraints that arise during the design process. This paper describes such a combined approach which is currently being used with two commercial CAD systems.

Keywords: Constraints, optimisation, computer aided design (CAD), mechanism design, machine design

1 INTRODUCTION

Computer aided design (CAD) systems have facilitated designer's task by automating many of the activities involved in a typical design process [1]. These systems were developed earlier in the sixties merely for the purpose of electronically archiving the paper based drawings and slowly evolved into 3D modelling tools that possessed the capabilities of assembly design and mechanism motion simulation. Nowadays these systems help designers in solving complicated engineering related tasks and can be integrated within a company's data management, planning and manufacturing facilities.

Earlier CAD systems acted as repositories for the geometric entities and had no interpretation by themselves for geometric representation. The advent of constraint based CAD has made these systems more intelligent [2] by enabling them to take some account of the meaning of the geometry. It also forms the basis for feature-based CAD modelling. In such a CAD system the components created are essentially parametric. This means the components can be reconstructed in different forms by simply changing the defining parameters. The relationships between individual geometric entities are defined using constraints. These constraints also define assembly and motion relationships. The aim is to resolve these constraints in order to create a valid assembly design. However the use of constraints in a typical CAD system is often restricted to the identification and resolution of geometric constraints. The underlying constraint resolver also takes the advantage that their form is known a priori. On the contrary, in a real design problem the constraints are present in various forms. These can result from any source, ranging from customer requirements to the manufacturing methods available. These types of constraints involve the design parameters but can also be dependent upon a particular design application. As an example consider the design of an electric motor. The motor should be powerful enough to produce the desired torque and yet should be small enough to fit into the available space. These types of design constraints are referred to as general constraints.

The another limitation of the CAD systems is that they cannot be customised for specific needs to incorporate general constraints as they are closed and it is difficult to know what is going on inside. This means that the designer cannot add further constraints into the system. This customisation is often necessary when the designer has to optimise a design subjected to general constraints such as cost and performance. Constraint based techniques on the other hand provide such flexibility of customisation

[3, 4]. These techniques have been proved useful in mechanism and machinery design. These techniques also have the advantage of solving constraint problems when their form is not known a priori. Using them in conjunction with the features offered by a typical CAD system can provide support for tackling higher and more general form of constraints that arise during the design process. This paper describes such a combined approach which is currently being used with two commercial CAD systems.

Section 2 describes a constraint-based modelling technique and also introduces a stand-alone constraint modelling system. Section 3 highlights the need for integration of the constraint-based modelling technique with commercial CAD systems. The methods of achieving such integrations with two commercial CAD systems are discussed in section 4. Section 5 provides two case study examples of modelling gripper and ejection mechanisms of a confectionery wrapping machine with the help of proposed integrated systems. Conclusions and future work is finally presented in the last section.

2 CONSTRAINT-BASED MODELLING

Constraint-based modelling involves identification and resolution of all the constraints known to be acting on a system. In this approach, all the design requirements are converted into constraints and the various design parameters that affect these are identified. Various procedures can be used to manipulate the parameters in order to resolve the constraints [5, 6], including graph-based searches and optimisation. This approach relies on 'truth maintenance' for obtaining solutions to constraint problems. Each constraint is essentially specified as an algebraic expression between certain of the design parameters. It is arranged that this expression is zero when it is true so that a non-zero value represents a measure of its falseness. When more than one constraint is present, their values are combined (as a sum of squares) and this value is here referred to as the overall "truth". When a set of constraints is to be resolved, first of all the overall truth of a problem is determined using the initial set of values provided with the variables. If the truth value is false (non-zero), the system tries to manipulate a set of declared variables using optimisation techniques [7] so that a truth value closer to zero is found. The use of optimization techniques has made it possible to allow the constraint rules to be formed into nested loops. This provides means of creating resolution processes that can maintain the integrity or operation of a design whilst allowing, at the same time, the ability to select and manipulate a set of 'free variables'. This in turn helps in the search of new solutions.

The approach is open so that users can add new constraints and relax existing ones when required. Hence it can be tailored to specific applications. The constraint environment discussed in the paper is essentially stand-alone and the interest is in transferring design data, together with assembly and simulation information into a conventional CAD system for visualisation and for more detailed design work.

2.1 Constraint modeller

The constraint modeller is software that incorporates constraint modelling techniques in its environment. It was developed to investigate the use of these techniques in the design of mechanism and machine systems [7]. This system underlies a user language, which acts as a normal programming language where variables can be declared, expressions between the variables can be evaluated and graphical entities can be displayed. It also allows the constraints between the design parameters to be specified and resolved.

In its basic form, this environment supports the creation of a number of three dimensional geometric entities such as points, lines and circular arcs. These entities can be grouped together in "model spaces", translated, rotated and scaled using transforms [8]. A model space provides a useful way to handle geometric entities. Every model space is associated with a transformation matrix that dictates how a geometric entity is mapped to a world space. These model spaces can be embedded into others which allow the formation of a model space hierarchy. In this way, a tree structure is obtained with its "root" as the world space. A partial assembly of the geometric entities is achieved using this concept.

A machine model in this system is created by defining different constraint rules. These rules also represent relationships between different design parameters. The constraint rules are setup as expressions between the variables. These expressions are deemed to be true when zero. Any non-zero value is the measure of falseness in the system. The system tries to resolve the constraint problem by changing the initial values of the free variables so that a solution closer to zero can be found. This resolution process is based upon the optimisation techniques [9]. It is also possible to identify such constrains that are not satisfied [7]. It gives the designer an option so that less important constraints can be relaxed and overall solution can be determined. A complete assembly is finally achieved using these rules.

As an example consider a four bar mechanism shown in Figure 1. In this example three links of the four-bar mechanism are represented as three lines La (driver link), Lb (coupler) and Lc (driver link) as shown in part (a) of the figure. These links are created within model spaces m1, m2 and m3 respectively. All the model spaces are initially embedded into world space (W). In order to achieve a partial assembly model space m2 is embedded into the model space m1. Hence when transformations are applied to m1, these are also reflected in m2. In other words coupler link (Lb) rotates with the driving link (La). This embedding gives the partial assembly shown in part (b) of the figure. Finally to complete the assembly, following constraint rule is used:

rule (Lb:e2 on Lc:e2) (1)

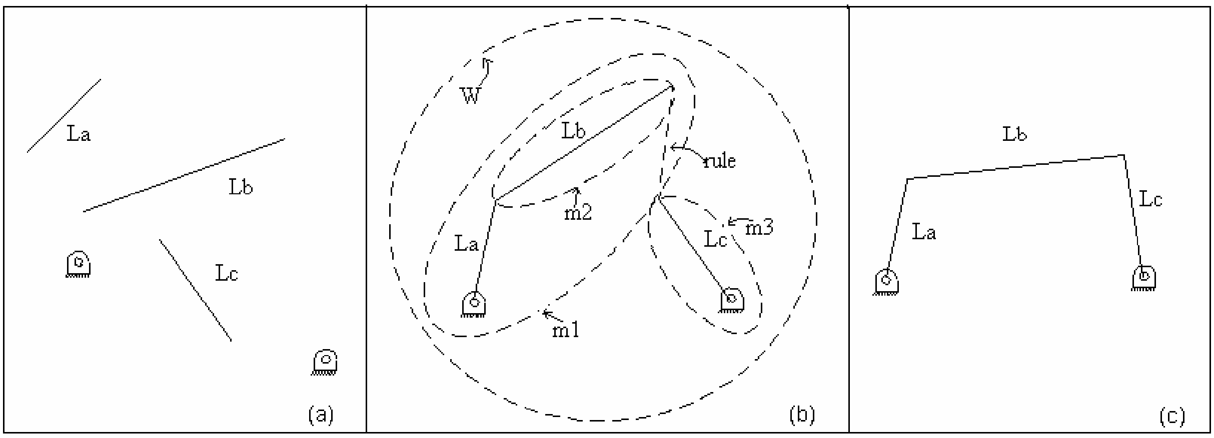


Figure 1. Assembly of a four-bar mechanism

This rule forces the free end point of link (Lc) to lie on the free end point of link (Lb). This is achieved by varying the rotations associated with their corresponding model spaces. Thus a required assembly is obtained as shown in the part (c) of Figure 1. The expression Lb:e2 in the underlying language represents the second end point of line one. The 'on' function returns the distance between two entities and so, in the rule statement is zero when the points coincide. Hence using the above method, end point of one line can be fixed to the end point of the second line. A 'pivot' function is another type of constraint available in the underlying language which restricts the rotation of an object only about an axis. A loop statement also exists that allows the geometric entities to be moved repeatedly to new positions. It helps in performing a simulation. In order to study the motion of this mechanism throughout its complete cycle, successive transformations are applied to the driver link (La) that in turn drives the whole mechanism.

3 NEED FOR CAD INTEGRATION

The constraint modeller is a useful tool for studying mechanism and machine systems. It can represent diverse types of machine components such as cams, linkages, rotary and linear actuators from the pneumatic, electrical and mechanical domains, within a single environment. This ability of the constraint modeller is useful for evaluating interactions between different machine components and thus a complete machine system and process can be studied. Some of the main features of this environment are listed below.

- The models constructed in this environment are driven purely by the constraints rules. Hence by simply inverting these rules one can also modify the input parameters. For example, in case of a cam and follower assembly, normally a cam profile is specified and then the resultant motion is analysed. However in the constraint modeller environment the desired motion can be specified and a cam profile can be created from it.
- The system incorporates a constraint resolution process which is based upon optimisation techniques. These techniques try to solve a problem by finding the minimum of the sum of squares of the constraint expressions. There is no need for the number of variables to be equal to

the number of equations to be solved. Thus optimization techniques naturally handle over- and under- constrained problems. This means that the system is well suited for solving problems whose form is not known a priori.

- The constraint modelling environment provides the flexibility of customisation. The designer can add further constraints depending upon the nature of the problem involved.

These features offered by the constraint modeller are normally absent in most CAD systems. These CAD systems, on the other hand, are highly suitable for 3D modelling, performing engineering analysis and generating manufacturing information for the designs. The main aim of the research work presented here is to create a protocol to enable transfer of design information from the constraint modeller environment to a commercial CAD system for viewing and for further design and analysis. This provides an interface where design activities such as modelling, assembly and simulation are controlled by the constraint system but are displayed via the CAD system. The basic design, once created using constraint modelling techniques, can be further analysed using advanced features within the CAD environment.

4 INTEGRATION WITH CAD SYSTEMS

The protocol is investigated with two commercial CAD environments representing two typical forms of such systems. One is Unigraphics NX3 which is an example of a "large" feature-based system. The other is Visual Components which is typical of emerging "smaller" systems. It is based on discrete modelling software and is proving useful in the design and simulation of manufacturing systems.

4.1 Integration with NX3

The NX3 software of Unigraphics is a commercial CAD system. It provides a designer with the capabilities of geometric modelling, design review and evaluation and drafting. NX3 comes with different modes of application. "Part modelling" is the most basic mode offered by this package which is helpful in creating 3D models of a design. Other applications, including assembly, motion, finite element analysis and manufacturing, are built over this basic mode.

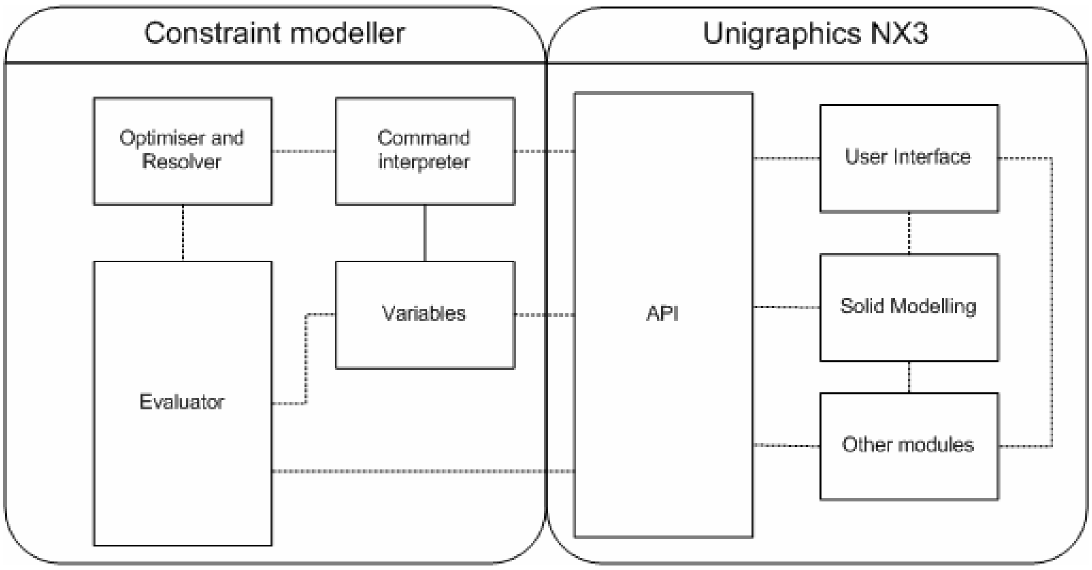
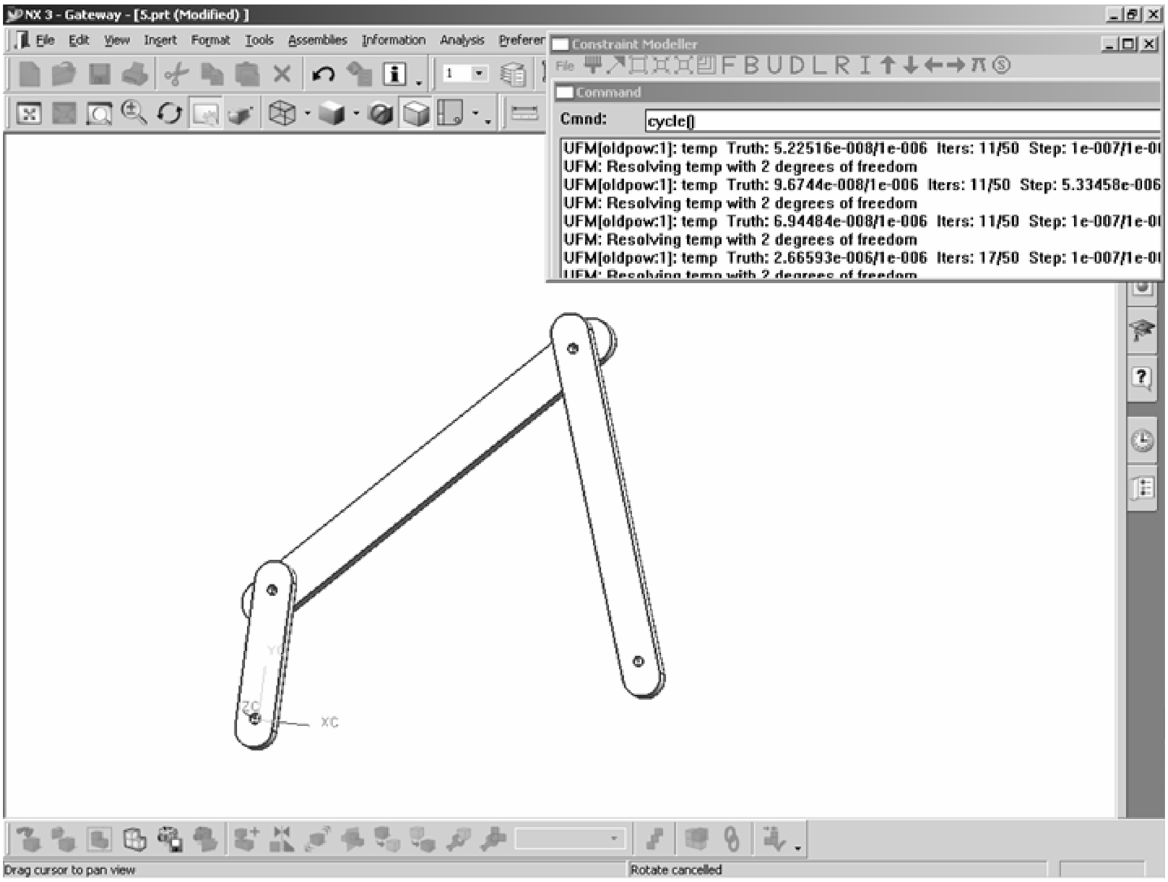


Figure 2. Structure of Constraint modeller - NX3 integrated system

The part modelling mode of NX3 has been used to integrate it with the constraint modeller. This integration is made possible with the help of Open API which is the application programming interface language of NX3. Open API provides the necessary routines, procedures, variables and data structures for the communication with different pieces of software and enables the integration of third party applications within NX3 environment. The structure of the integrated system is shown in Figure 2.

In the integrated system, the user interface and display is provided by the CAD system itself. Commands for the constraint modeller are received by the command interpreter via a command window. The purpose of the command interpreter is to create and manipulate user variables. It also evaluates the expressions involving variable which may also include geometric objects. The constraint resolver deals with the constraint expressions and works directly with the evaluator to try to optimise the constraint expressions to be zero.

The command handling and constraint resolving parts of the constraint modeller operate as previously. The difference lies in the way geometry is created within the integrated system. The appropriate Open API calls are made which create the required objects in the CAD environment. A pointer to the geometric object is returned by the CAD system after its creation. This pointer is held as the "value" of the corresponding design variable within the constraint modeller so that it can be subsequently referenced.



Cmnd: cycle()

```
UFM[oldpow:1]: temp  Truth: 5.22516e-008/1e-006  Iters: 11/50  Step: 1e-007/1e-0
UFM: Resolving temp with 2 degrees of freedom
UFM[oldpow:1]: temp  Truth: 9.6744e-008/1e-006  Iters: 11/50  Step: 5.33458e-006
UFM: Resolving temp with 2 degrees of freedom
UFM[oldpow:1]: temp  Truth: 6.94484e-008/1e-006  Iters: 11/50  Step: 1e-007/1e-0
UFM: Resolving temp with 2 degrees of freedom
UFM[oldpow:1]: temp  Truth: 2.66593e-006/1e-006  Iters: 17/50  Step: 1e-007/1e-0
UFM: Resolving temp with 2 degrees of freedom
```

Drag cursor to pan view

Rotate cancelled

Figure 3. Assembly of a four-bar mechanism

The main advantage of the integrated system is in creation of assemblies and motion simulations that are controlled externally by the constraint modeller. The constraint modeller calculates the necessary transform matrices for the objects and passes them, with the corresponding object pointers, to the CAD system. Again this is accomplished using appropriate Open API procedures. It also enables the creation of model space hierarchies and hence a successful assembly is achieved. Using this, other constraints such as kinematic relations in terms of velocity and acceleration, can be imposed which help to avoid clashes and investigate possible failure conditions. Figure 3 shows a screen shot of the integrated system. A simple four bar mechanism that was modelled, assembled and run using the integrated system is shown. The additional window on the top right corner is the window for constraint modelling commands.

4.2 Integration with Visual Components

Visual Components is a combined CAD and simulation package. This system supports 3D modelling by providing means for the creation and manipulation of 3D geometry. One important feature of the Visual Components system is the existence of a stand-alone "viewer" (3D Video). This allows a design to be visualised and, in the case of kinematic system, an existing simulation to be re-run without the need for the full version of the software. The created simulations are compact, reusable and can be viewed from any orientation while they are running. A geometric optimisation feature of the system optimises CAD geometry by removing un-necessary details (such as simplification of fillets and rounds) and thus reduces the size of the simulation. These features of the system open up the possibility of performing initial constraint-based design and optimisation as a “node” within a distributed design environment. Here the outline design requirements are supplied (possibly in text form) to the node. The appropriate constraint-based formulation is set up and investigated. Designs are then investigated and can be transmitted to the other distributed nodes for visualisation (via the viewer), comment and further analysis (Figure 4). Once the outline design is finalised, the design information can be passed on to other nodes for further design work.

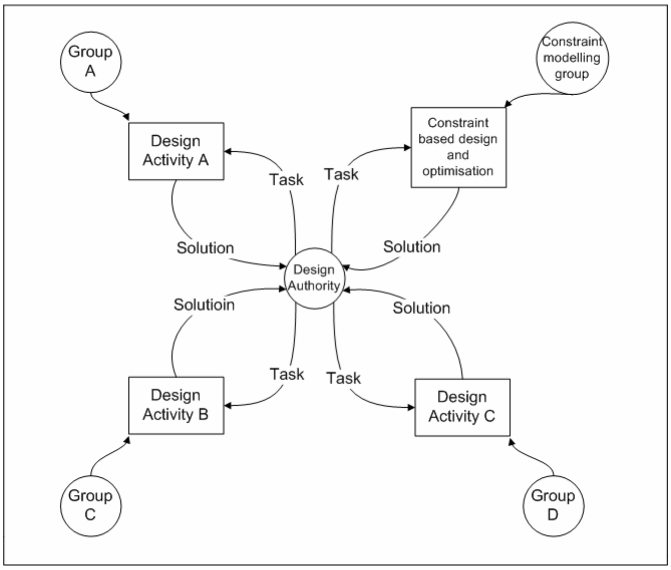


Figure 4. Example of a distributed design environment

The interaction with Visual Components is achieved via the use of that system’s command language which is implemented via the Python scripting language [10]. Here the design and assembly information within the constraint environment is used to create an appropriate script. This then enables it to be recreated within the Visual Components system.

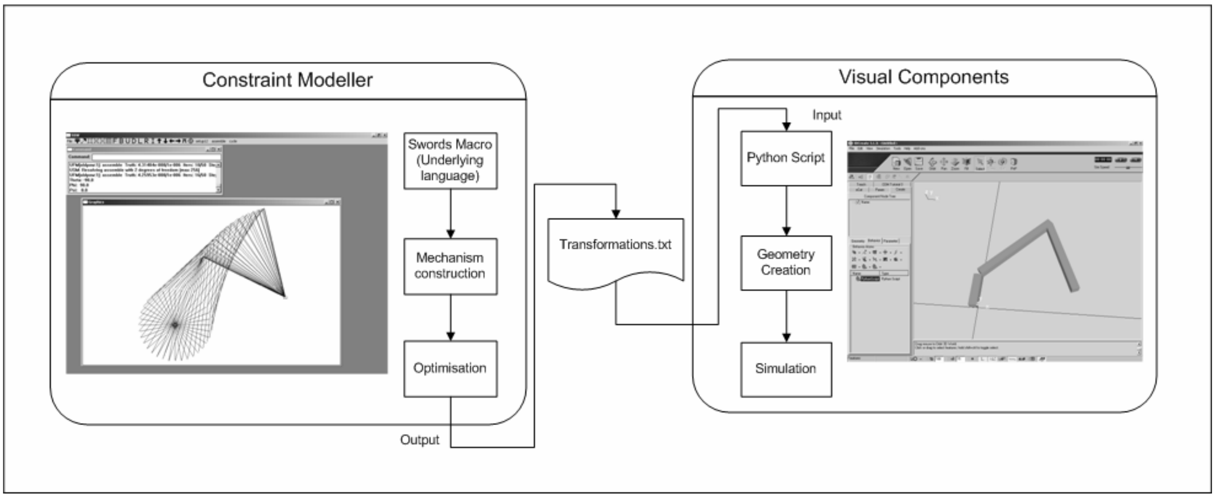


Figure 5. Structure of Constraint modeller - Visual Components integrated system

Figure 5 shows the structure of the interfaced system (constraint modeller and Visual Components). A mechanism model is first created and optimised within the constraint modeller environment. An output script file is then generated from this environment. This file contains all the information regarding geometry, assembly and operation of the mechanism. This information is stored in terms of spatial transformations of various parts of the mechanism. The script file is read into Visual Components environment using the Python script language. This then recreates the mechanism model and runs it as a simulation (in the case of a kinematic system). The geometry of the model is either created within

Visual Components using the Python script or it can be imported as external CAD geometry. After geometry creation, transformations are applied to it in order to assemble the component parts into their correct global positions and finally a mechanism simulation is generated.
The modelling of a four-bar chain example with this integrated system is shown in Figure 5. Initially a model of four-bar chain is generated, run and optimised within the constraint modeller environment as shown in left side of the Figure 5. The output script file is then generated from the constraint modeller environment. This file is then read into the Visual Components environment using the Python script language. The three links are generated by the script language itself as it invokes the underlying command language of the system (shown in the right half of the Figure 5). Then transformations are applied to respective links and a simulation is generated using this interface.

5 CASE STUDY EXAMPLES

This section presents modelling of two case study examples with the integrated systems. The mechanisms investigated here are parts of a confectionery wrapping machine. The two mechanisms considered are; a gripper mechanism and an ejector mechanism. The purpose of the gripper mechanism is to pull the wrapping film from a reel. The film is cut off, once a sufficient amount has been withdrawn. It is then positioned above the sweet to be wrapped. A separate mechanism moves the sweet vertically upwards into the film to start the wrapping process. Once positioned correctly, two rotary driven gripper jaws transfer the wrapped sweet to the ejection station. The function of the ejector mechanism is to push the wrapped sweet from the transfer grippers onto a chute. Here the wrapped sweet finally exits the machine.

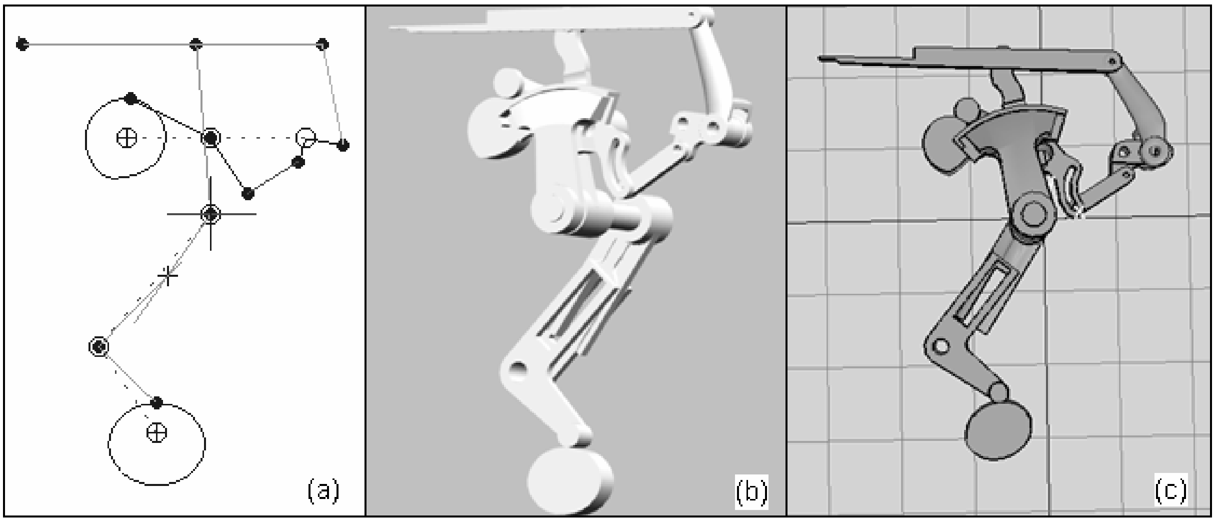


Figure 6. Gripper mechanism

Figure 6 shows the gripper mechanism of the sweet wrapping machine. Part (a) of the figure shows a “stick diagram” representation of the mechanism. This was modelled within the constraint modeller environment. Here different lines represent various links of the mechanism. The closed curves are the representation of two driving cams. The purpose of these cams is to control the horizontal and vertical motion of the mechanism. In this environment each link is embedded into its respective model space. A partial assembly is obtained by forming the hierarchy of model spaces. Assembly is completed by imposing constraints so that the cam follower links always lie on the appropriate cam profiles. The following rules are used to achieve this.

rule(camfollower1:e1 on cam1) (2)

rule(camfollower2:e2 on cam2) (3)

Part (b) of the Figure 6 shows the assembly of gripper mechanism modelled in the constraint modeller-NX3 integrated system. Here various links are modelled using NX3. The assembly of these is achieved using the model space hierarchy and the constraints. In this system the constraint modeller directly controls the transformations and the assembly of various links. Once a successful assembly is achieved, its motion can be simulated by allowing the cams to rotate in steps and resolving the constraints at each stage.

Part (c) of the Figure 6 shows the same mechanism in the constraint modeller-Visual Components integrated system. Again the geometry of the links is created in Visual Components itself and the constraint modeller provides the transforms for the assembly and simulations.

This simulation of the gripper mechanism allows the track of the end of the gripper to be investigated. An advantage of these systems is that once an assembly of a mechanism is generated, the user can always invert the constraint rules and specify the desired output motion which then generates the required input motion. In this case the input motion is determined by the profile of the two cams. If any modification is done to the output motion of the gripper mechanism, the system automatically updates the required cam profile.

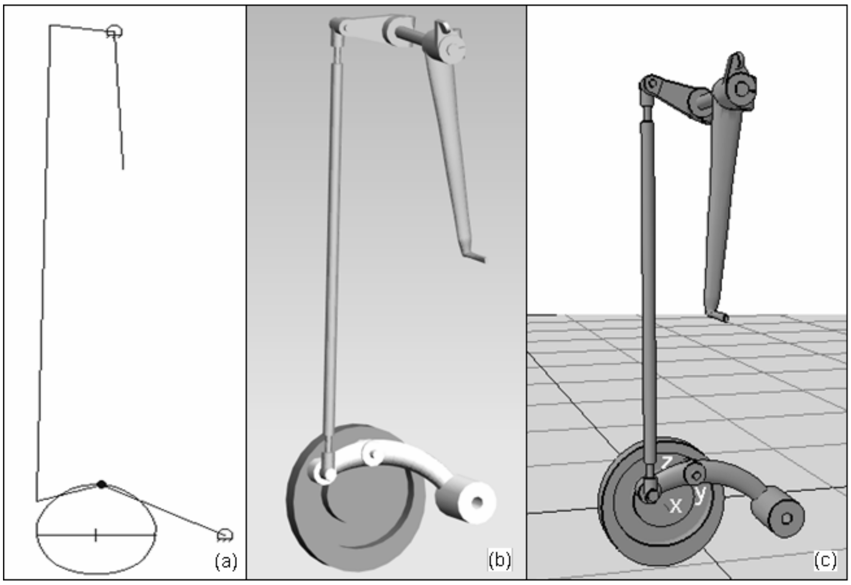


Figure 7. Ejector mechanism

Figure 7 shows the assembly of the ejector mechanism. Part (a) of the figure shows the "stick diagram" representation of the mechanism. This is modelled in the constraint modeller environment using the same concept explained earlier. The ejector arm oscillates horizontally and its motion is controlled by a cam. Parts (b) and (c) shows the same mechanism modelled in the constraint modeller-NX3 and constraint modeller-Visual Components integrated systems respectively. The movement in this case is also controlled purely with constraints. Any required output motion can generated from the mechanism by simply forcing the ejector arm to follow the prescribed path. This in turn improves the cam profile that defines the input motion.

6 FUTURE DEVELOPMENTS

The previous case study examples show the application of the approach to simple problems. However it is the aim of the research to extend the methodology to allow large, complex and real problems to be handled. One extension proposed is to incorporate constraint modeller within the chosen CAD systems to allow constraint based operations to be performed in more than one distributed tasks. Whilst this requires greater knowledge of constraint resolution and optimisations techniques it does allow decisions to be made between the various tasks at a higher level that could include the combining of the rules from individual tasks to allow 'super-tasks' to be formed.

Another variation of this distributed approach being considered is one in which multiple problems are formed and their interactions resolved. Current research is being undertaken into the issues of health and safety of machines within a manufacturing environment. In order to study the machine and the human interaction, models have to be created of both in operation. Whilst the desired operation of the machines can be identified alone, modelling of the actions of the human operator have not only to include knowledge of the tasks to be carried out but also needs an understanding of the actions, posture and stability of the operator. Once these are understood the interactions can be determined, and whether these constitute a dangerous situation can be determined.

6 CONCLUSIONS

CAD systems are helpful in facilitating a designer's tasks in many design activities. The features offered by these systems can be augmented by the use of constraints based procedures. However the application of the constraints in most of the CAD systems is limited to geometric entities. There is still a need for dealing with the constraints that express the general functionality of a design. This paper introduces an approach based upon a constraint based modelling technique integrated with a CAD system. Two commercial CAD systems have been investigated for this integration purpose and a protocol was created to transfer design information from the constraint modeller to the CAD system. The integrated system presents a constraint based approach to achieve assembly and motion simulations. In such a system a design is constructed with the help of constraint rules. These rules can be altered, weighted and inverted at any stage to improve the basic design. The designer has the advantage of being able to add new constraints to the system without modifying the underlying structure of the system. The system also provides better control of the assembly to the designer and thus enables the investigation of different arrangements. This approach has been demonstrated with two case study examples that involve modelling, assembling and simulation of two mechanisms with a confectionery wrapping machine. It has allowed the motion of various parts of these mechanisms to be investigated and improved.

ACKNOWLEDGEMENT

The current research work is carried out with the support of a DEFRA research grant given as part of the Food Processing Faraday-KTN. This support is gratefully acknowledged together with that of other collaborators

REFERENCES

[1] Cederfeldt, M. and Elegh, F. Design Automation in SMEs - Current state, potential, need and requirements. In *Intl Conference on Engineering Design*, *ICED'05*, Melbourne, Aug. 2005.

[2] Brix, T., Brüderlin, B., Döring, U., and Höhne, G. Feature- And Constraint-Based design of solution principles, In *Int. Conference on Engineering Design, ICED '01*, Glasgow, Aug. 2001, pp. 613-620.

[3] Markus, A., Vancza, J., and Kovacs, A. Constraint-based process planning in sheet metal bending. *CIRP Annals-Manufacturing Technology*, 2002, 51(1), 425-428.

[4] Schmidt, L. C., Shi, H., and Kerkar, S. A constraint satisfaction problem approach linking function and grammar-based design generation to assembly. *Journal of Mechanical Design*, 2005, 127(2), 196-205.

[5] Anantha, R., Kramer, G. A., and Crawford, R. H. Assembly modelling by geometric constraint satisfaction, *Computer-Aided Design*, 1996, 28(9), 707-722.

[6] Hicks, B. J., Medland, A. J. and Mullineux, G. The representation and handling of constraints for the design, analysis and optimization of high speed machinery, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing, AIEDAM,* 2006, 20, 313-328.

[7] Mullineux, G. Constraint resolution using optimisation techniques, *Computers & Graphics*, 2001, 25(3), 483-492.

[8] Leigh, R. D., Medland A.J., Mullineux, G., and Potts, I. R. Model spaces and their use in mechanism simulation, *Proc IMechE Part B Journal of Engineering Manufacture*, 1989, 203, 167-174.

[9] Ge, J.-X., Shang-Ching, C., and Gao, X.-S. Geometric constraint satisfaction using optimization methods, *Computer-Aided Design*, 1999, 31(14), 867-879.

[10] Martelli, A. *Python in a Nutshell,* O'Reilly, Sebastopol, CA, 2003.

Contact: B. Singh
University of Bath
Department of Mechanical Engineering
Bath, BA2 7AY
United Kingdom
Tel: (00) 44 1225 385937
Fax: (00) 44 1225 386928
b.singh@bath.ac.uk