Design and Modeling of a Mechatronic Packaging Machine

D. Borghi*, R.Borsari*, M.Poppi*, E. Sacchetti* and L. Bassi**

* Tetra Pak Carton Ambient S.p.A., Modena, Italy

** University of Bologna, Department of Electronics, Computer Science and Systems, Bologna, Italy

***Abstract*— In this paper we present a model-based systems engineering approach to the development of mechatronic systems, whose main characteristics are:**

- **A strong attribute/requirement driven structure**
- **The central role played by the functional decomposition of the process being implemented by the system under development**
- **The extensive use of modeling and simulation to continuously support all the phases of system development**

Major advantages of the proposed approach are: the strong decisions basis given by the extensive use of simulation models from the early phases of the development effort, the full traceability of design decisions to corresponding requirements and verification/validation activities, the possible re-use of existing already developed architectures/solutions as well as simulation models, and fast quantitative evaluation of different proposed alternatives. Furthermore the systems engineering perspective enables the early consideration of non-functional attributes such as costs, reliability, manufacturability and so on.

The proposed methodology is illustrated with a simplified example taken from an industrial application: the development of the drive unit of the package-forming module of a filling machine for liquid food packaging.

I. Introduction

Engineering design of systems is nowadays showing a strong tendency towards the creation of more and more complex products encompassing several different engineering domains. Mechatronic systems, integrating together mechanical, electrical/electronic (and possibly hydraulic/pneumatic, etc.) components with information processing and control elements are a typical example of this class of products. On the other hand we assist also to an increased pressure on shortening development lead times and cutting costs.

Managing these somehow conflicting requirements is a formidable task for modern product development processes. System modeling and simulation, Computer Aided Engineering and Virtual Prototyping are today widespread technologies that may help reaching the aforementioned goals while keeping an increased trend in product quality control. Several tools implementing different capabilities are now commercially available and start to have a quite large diffusion among industrial development communities. Furthermore system verification/validation (at least partially) by computer modeling and simulation is becoming a common practice, especially in high technology industrial segments like automotive and aerospace. However it is usually not enough implementing some more or less advanced simulation tools in order to completely enjoy the potential benefits these new "virtual" technologies; they instead should be included as tools into a well defined and structured design process, specifically addressing all the issues introduced by the aforementioned increased complexity of the products to be developed.

Many different approaches to the definition of the design process exist in literature: Axiomatic Design [1], Design Process Roadmap [2], Theory of Domains [3], Total Design [4] and Systematic Engineering Design Process [5] are just some examples of well structured proposals that take very different perspectives. Most of the existing attempts, although quite general and abstract and, as such, widely applicable, more or less explicitly refer to traditional (single engineering domain) design, with little effort spent on the problems related to the synergetic integration of technologies coming from different fields and to their control. Some relevant exceptions started to appear in literature in the last decade trying to merge together traditional established practices with modern systems engineering approaches. The systems engineering perspective, focusing on the rational management of well defined requirements structures clearly defining the design problem to be solved and taking into consideration the multidisciplinary aspects of the development by the introduction of solution architectures [6], enables quite naturally the holistic integration of multidisciplinary technical solutions. Among others we cite the guideline VDI 2206 [7] as a quite successful attempt to cover all the relevant methodological aspects of the design of mechatronic systems.

The implementation of this kind of design processes into the industrial practice of packaging systems development is not straightforward and needs some additional logical steps. Indeed packaging systems involve the synergetic cooperation of many different complex processes that must be taken into consideration both for the correct definition of the requirement structure and the development of solution architectures. Starting from the functional decomposition of the package forming process, we propose a model based system engineering approach to the development of the mechatronic systems that are supposed to implement the identified functions. The detailed functional analysis of the process enables a correct identification of the main attributes of the process and the corresponding value ranges, giving rise to a well structured functional requirement definition. This structure can then be easily used to trace design decisions to the corresponding requirements and functions, establishing the basis for the verification and validation processes. This can be accomplished by supporting the requirements management process continuously with the creation of suitable simulation models that capture the essence of the described functions and provide quantitative information for decision bases. Furthermore the adopted systems engineering approach allows the early introduction into the design process of non functional attributes like cost, reliability and manufacturability in a fairly straightforward and natural way enriching the aforementioned decision basis with different and wider perspectives.

The paper is organized as follows: in the next section we first briefly describe the package forming process typical of Tetra Pak Carton Ambient filling equipments. We then introduce our approach to process functional decomposition and integrate the resulting process description into a proposed design process model. To illustrate the concept a simplified example describing the main steps in the design of a mechatronic drive for a package forming unit then follows in section 4. We then try to draw some conclusions and line out possible future developments in section 5.

II. The Package Forming Process

Tetra Pak develops, produces and markets food processing and packaging systems, i.e. processing equipments, packaging materials, package specifications, packaging machines and distribution equipments. Tetra Pak Carton Ambient more specifically is mainly involved in aseptic packaging systems and this means that strong focus is on process hygiene and integrity of the packages, in addition to other attributes, common to other types of packaging systems, like package appearance, cost, convenience and so on. In order to fulfill the strong quality/legal requirements typical of aseptic packaging, an architectural choice for the filling machine has been taken: we develop roll-fed (in contrast to blank-fed) systems. The resulting process is schematically shown in fig.1.

The roll-tube concept enables space savings before and after filling and high confidence that the whole surface of the packaging material is correctly sterilized, since planar surfaces fulfill the requirements of an aseptic system, in terms of time/temperature exposition to sterilization agents and successive drying, more than complex geometries. Furthermore, the complexity of the filling system is greatly reduced with increased hygienic level and lastly, the resulting packages are totally filled, enabling high product quality and good distribution properties.

In order to manage the extreme complexity of the package forming process, a functional decomposition approach has been taken [8], treating the other concurrent processes (aseptic, filling and sealing) as external processes connected through interfaces. This allows the development of a functional model of the forming process based on two main concepts [9] [10]: function, defined as a performed operation that is expressed as the active verb of a verb-object pair, and flow, defined as the change with respect to time of the object of the verb-object pair. A flow is the recipient of the function operation. Three types of flows are used: material, energy and signal, even if the distinction is purely a matter of convenience since all signals are in fact energy flows.

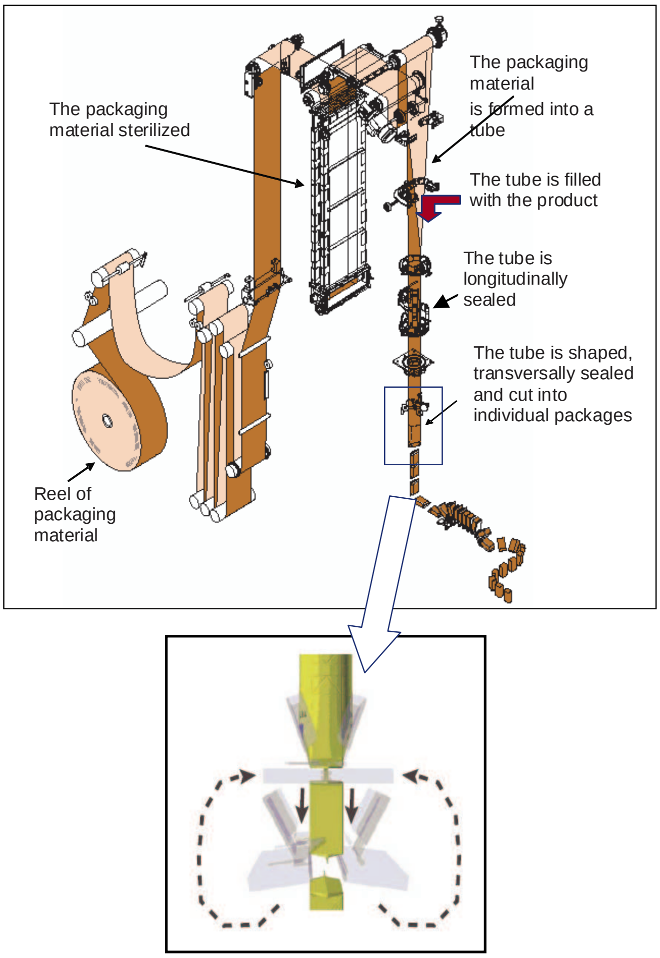

Fig.1 The package forming process

The functional network resulting from the analysis is a quite abstract, but very useful description of the process in terms of the elementary functions that are required to achieve its overall purpose. The complete functional network of the forming process is too complex to be shown and described here, but as an example in fig.2 we present a simplified chunk of the network with a single function shown with all its inputs and outputs. The complete network is made of about forty functions.

The approach followed during the development of the functional model has been to take as much as possible a solution independent decomposition, identifying only the functions that are strictly needed to transform the reel

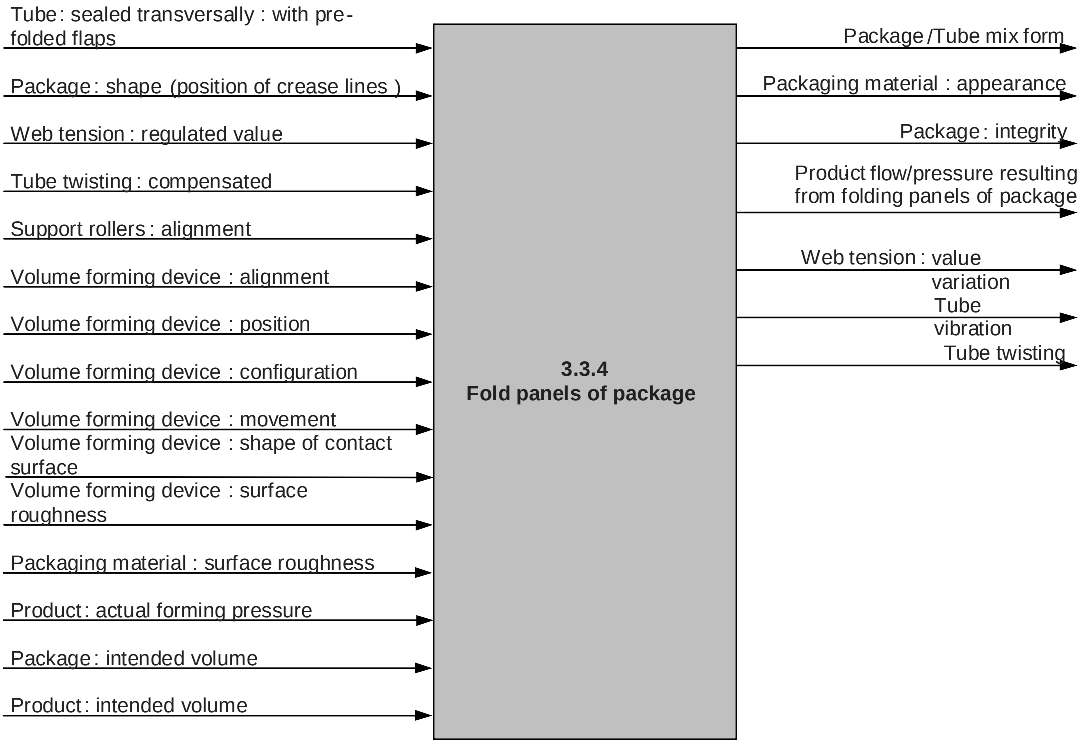

Fig.2 Functional network – single function example

of packaging material into fully filled, tight sealed packages, as seen from the packaging material perspective, leaving aside the other functions added by the specific devices used to implement the process (which instead are of course solution dependent).

The availability of the functional network opens several different possibilities: first of all it is easy to follow the main flows of materials, energy and information along the process, clarifying the relevant changes of state. Secondly the main input/outputs variables of each function can be identified, classified and qualified/quantified according to the specific function description. This is accomplished either by defining empirical relations based on heuristics or experience or using quantitative data coming from standard test methods or simulation models.

The complete collection of the function descriptions, the functional network and the relationships between the process variables constitute the pillars of our forming process model, which may be used to relate process inputs to outputs, to identify the main attributes and define the relevant functional requirements for the design of forming units. In fig.3 we summarize the relationship among process inputs, outputs and the model.

In the next section we will show how the described process model may be integrated into the design process.

III. The Design Process

The definition of a model for the design process is a formidable task, and there is no universally accepted such model even inside the academic community. Fortunately for our purposes it is enough to define just some basic principles that may drive design activities without specifying their details.

Usually a development effort starts with some kind of identification of needs that may come directly from the market, or from some specific company strategic decisions. Taking this perspective, generally speaking the design process may be defined as the iterative process to transform the statement of needs to product specifications and may be split into two main activities: problem definition and problem solution. Problem definition is covered by the so called requirements engineering process, while problem solution instead involves the solution definition and the solution evaluation sub-processes. To these processes it should be added some kind of decision analysis process as well.

The starting point is the collection of the customer and, more in general, of stakeholders' requirements about the system being developed. These define the problem needing solution and the context and external systems with which the system will interact. All this information is usually included into the "Originating Requirements Document" [6]. Through a Quality Function Deployment process these originating requirements are then translated into engineering or system requirements, which are stated in a more technical language and define target values for a number of attributes that will be used to qualify the design during its development. The resulting "System

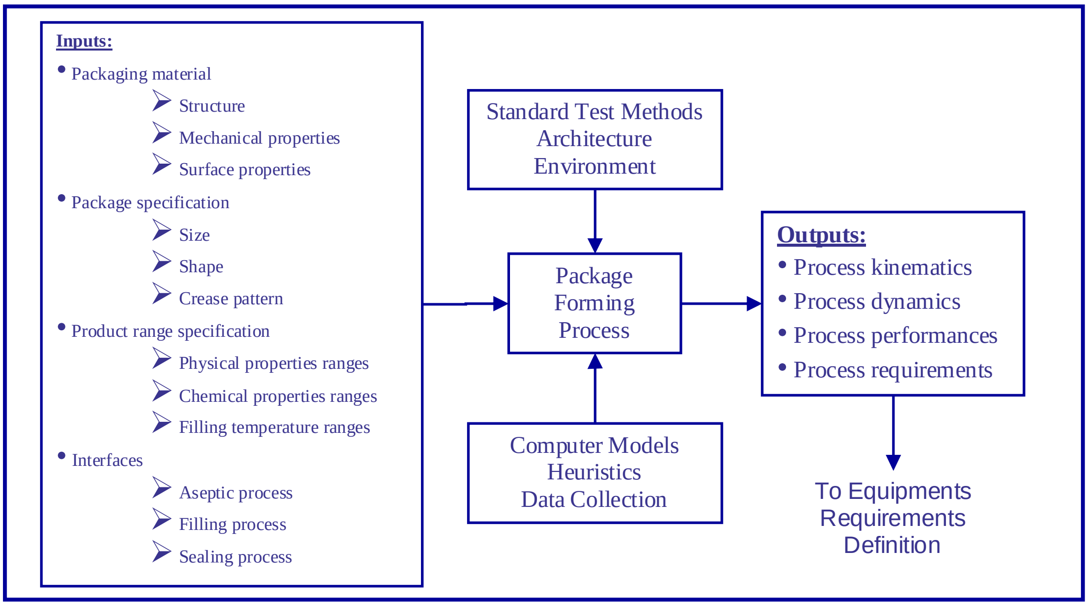

Fig.3 The Package Forming Process

Requirements Document" is the core enabler of the requirement-driven development process. Indeed system level requirements are then used to generate lower level requirements for sub-systems, assemblies, and components up to each configuration item. The corresponding evolution of the system requirement document is accordingly used as the base document to define the design activities relevant to each phase of the development process. Modelling and simulation are fundamental tools either for the verification/validation of requirements and for the generation of new requirements derived from the existing ones.

This requirement flow down process is fundamental to ensure full traceability. In this way a clear qualification strategy can developed at each stage of the development process, and if the process is correctly implemented, virtually no originating requirement should remain unanswered, resulting in a better overall quality control on the developed product.

The requirement flow down process is usually accomplished through modeling and analysis and is based on three main techniques [6]:

- Apportionment spreads a higher-level requirement into lower level requirements maintaining the same units. Typical examples are cost, reliability, durability and so on.
- Equivalence is a simple flow down technique that causes the lower level requirement to be the same as the higher-level originating requirement. This technique usually is most appropriate for constraints (e.g., "all materials interacting with the product must be approved by the authorities").
- Synthesis addresses the situation in which the higher-level requirement is comprised of complex contributions from lower-level requirements and usually requires the development of some analytical models. This technique is quite common for functional requirements.

Integrating the previously described forming process functional model and the requirements engineering process allows us to describe the proposed design process that is based on the standard V model of product development processes (see fig.4).

We start from the originating requirements set to define the main inputs of the required forming process. The analysis of the forming process will provide basic functional system requirements and will support the creation of models. The requirement flow down process then proceeds until the complete definition of the design problem is reached. It is worth to note here that the process is not strictly sequential, on the other hand usually several loops are needed to complete the task. Secondly, during the requirement engineering process a lot of design work is actually done, since the creation of several models of different complexity and level of detail not only clarifies the design tasks, but also discovers several concepts that may be evaluated on the basis of the quantitative analysis supported by the same models. Furthermore, although not explicitly shown, requirements engineering involves from the very beginning an interdisciplinary approach that does not discriminate or separate different engineering fields.

Once the requirements flow down process reaches the point where requirements are specified up to the component level, after a selection process taking advantage of the developed models, the selected concepts

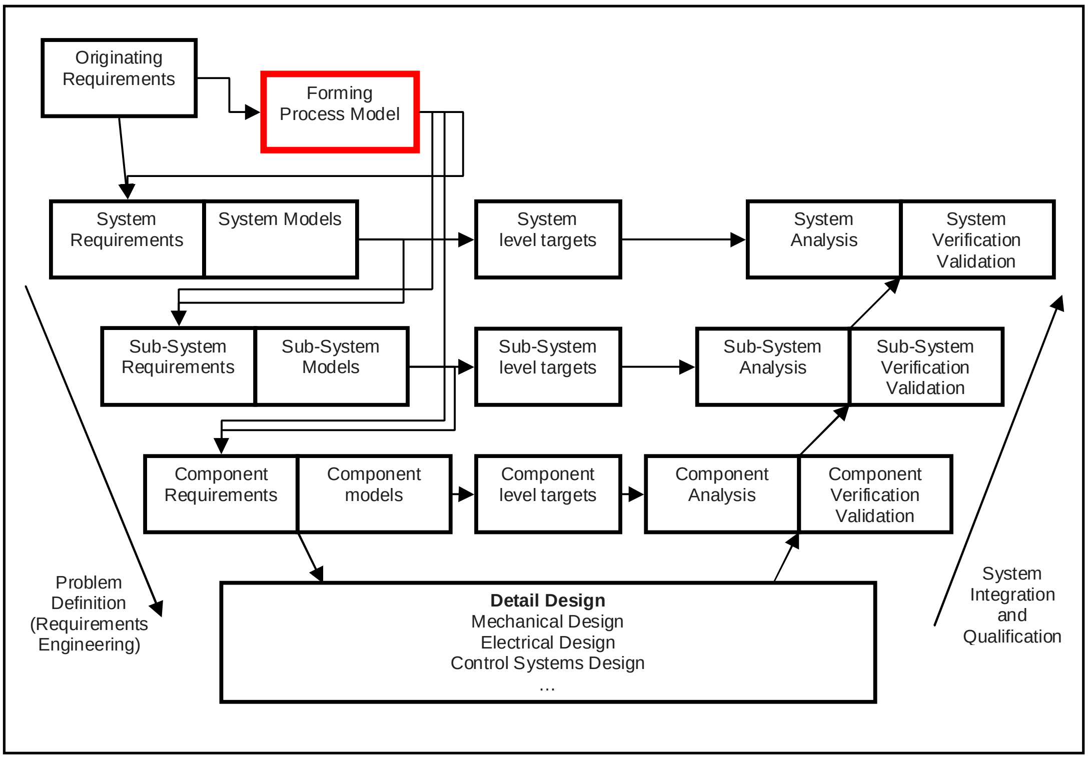

Fig. 4 Design Process V-Model

may go to the detail design stage, where each configuration item is completed. The next stage is the system integration phase where configuration items are assembled together. Again the defined requirement structure and the corresponding models may support verification/validation at different levels of integration. We note here however that climbing the ascending branch of the V-Model, simulation models development usually becomes more and more complex since it may involve a truly multi-paradigm approach including multiple physical filed coupling, different spatial-temporal scales and possibly different states of the matter, and from a practical point of view it may not be the most effective approach to sub-system/system verification/validation.

IV. The Design Of A Package Forming Unit

Originating requirements for a new package forming unit are usually expressed assigning targets to some established set of attributes such as capacity, operational cost, flexibility, reliability and so on. Starting from this point it is necessary to perform a first draft forming process design, in order to clearly determine the functional requirements that the unit will have to fulfill. Using our functional network and related qualitative/quantitative models it is possible at this stage, starting from the assigned originating requirements, to define process kinematics, i.e. the temporal sequence of deformation (geometry and motion) steps needed to transform the tube of packaging material into filled, tight sealed packages fulfilling shape and size requirements. Process kinematics enables the definition of the trajectory and the shape of the tools that mechanically interacting with the packaging material, translating some of the originating requirements into system requirements expressed into more familiar engineering terms. Indeed the next step in the design process is to synthesize a suitable mechanical system that is able to fulfill these requirements set. Typically common engineering requirements in mechanism synthesis may be classified as follows [11]:

- Topological requirements specifying the nature of the motion and the degrees of freedom
- Functional requirements specifying the number of independent outputs, the task to be accomplished by each output and the complexity of each task
- Constraints such as dimensional or inertial constraints.

Once these system level requirements are defined, they are translated into functions that are in turn mapped into suitable physical architectures (structured collection of elements which realize the specified functions). These architectures are then evaluated based on technical and economic criteria to determine suitable concepts. The resulting kinematic synthesis problem, usually very complex to solve, is simplified in our case since we have a catalogue of solutions already available among which we can select the one that best suites our purposes. In our example we develop a system that fulfills flexibility

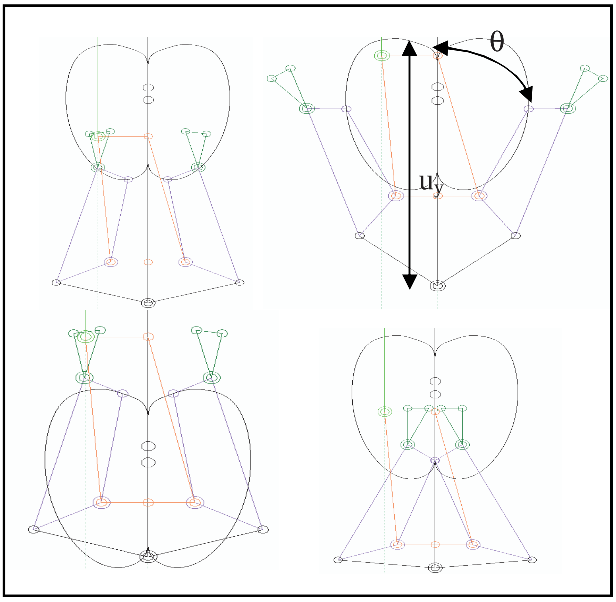

Fig.5 Selected Mechanism and Tools Trajectories

requirements (other architectures exist that fulfill different kind of requirements, such as high capacity [12] or low cost). This means that the resulting solution must allow change of volume/format/shape of the package with minimal changes in machine components, in a short time frame, without losing sterile settings. Furthermore, to allow flexibility, the forming unit must allow fast change of the trajectory of the tools interacting with the packaging material tube. Among different possibilities, to fulfill these functional characteristics, we select a “scissors-like” mechanism made of a slider crank mounted on a translating member. Given the mechanism topology and structure, the synthesis problem is reduced to finding the optimal dimensions of the links that enables precise function generation according to the requirements of forming process kinematics. This can be easily accomplished by constrained optimization analysis of a simple kinematic multi-body model. The skeleton of the resulting mechanism is shown in fig.5 in four different configurations along the required trajectory to be followed.

The mechanism has two degrees of freedom, that from a conceptual perspective are the displacement u_y and the angle θ in fig.5.

We are now ready to start a preliminary embodiment of the selected solution. This includes: the definition of the assembly structure and the preliminary part shapes, a preliminary materials selection and inertial properties evaluation. The characteristics of the motion profiles to be generated, and the inertial properties estimation allow also the definition of the requirements for the drive unit.

In order to fulfill also trajectory flexibility requirements we select a mechatronic servo drive architecture shown in fig.6. The motion profile generator is executed in the controller and generates the set point for the drive that in turns executes position, speed, torque loops. Finally the power electronics transform the values into physical entities (i.e. Ampere). The feedback system (resolver or encoder for position and velocity, Hall effect sensor for current) closes all the three loops. This servo drive architecture allows high kinematics performances in terms of: strict profile following, high system bandwidth, system stiffness and, of course, possibility to change profiles at will.

The two mechanism degrees of freedom are then driven by two translational axes moved by two AC servomotors with timing belt/pulley couplings. In order to form packages we need two such mechanisms with identical motions shifted in time by half period. To verify that this type of solution fulfills the functional requirements set by the process, we use different models of increasing complexity reflecting the different confidence level about the system under development acquired during the design process. Initially we have to dimension the belt/pulley drive according to kinematic/dynamic requirements. Then we have to investigate that the solution does not introduce unexpected vibrations into the drive due to flexibility of the couplings and that the position errors due to the elasticity of the belt is within the limits set by the process model. A model of the belt/pulley drive where the belt elasticity is lumped into variable spring stiffness components [13], together with variation of the natural frequencies of all the belt drives as a function of machine degrees (non dimensional time scale).

A more detailed model, using a mixed multibody/finite elements approach to account for distributed elasticity, is then used to understand how dynamically the elasticity of the belt affects the precision of motion transmission of the motion.

Furthermore, a model able to reproduce the behavior of the chain composed by the controller, drive,

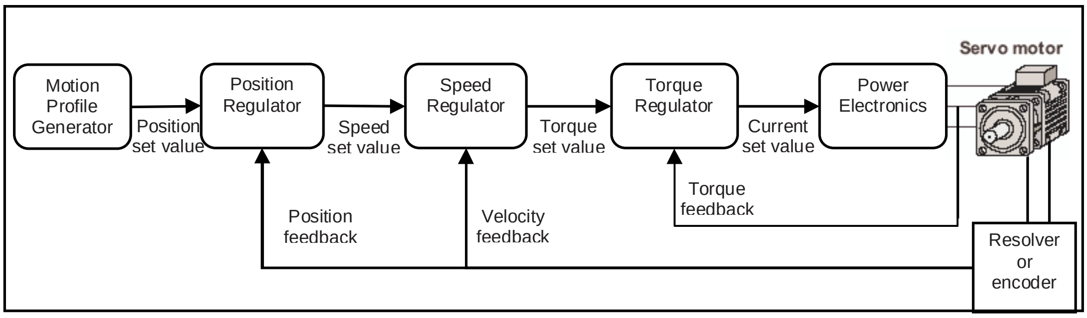

Fig. 6 Drive Unit Architecture

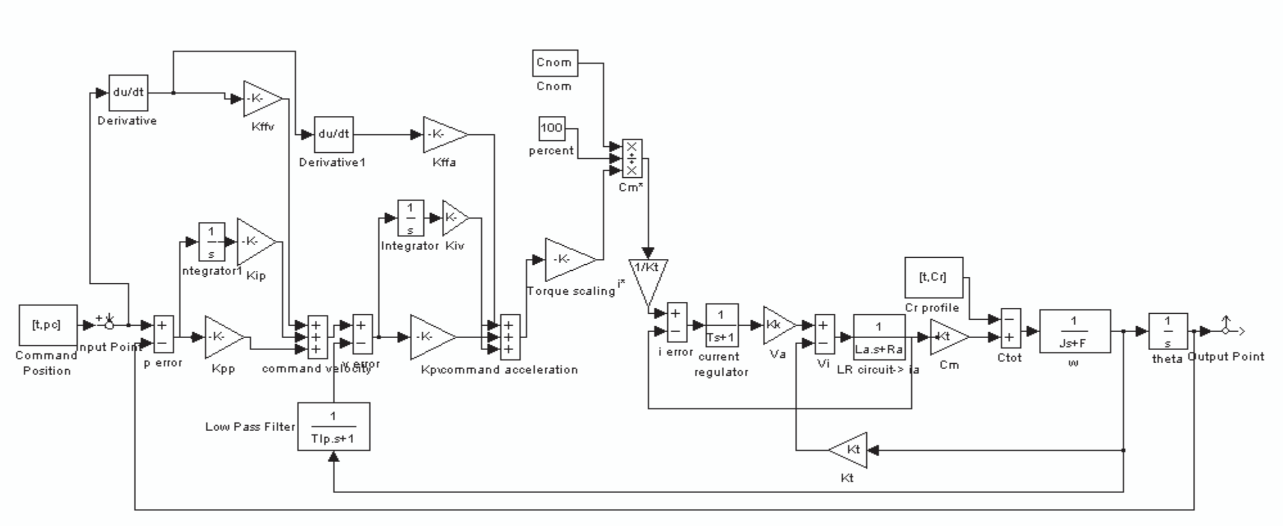

Fig. 7 Model of controller, motor and load

motor and load is used [14] to analyze the connection between the zeros/poles of the system and the various parameters of the adopted PID control.

The model shown in fig.7 has been realized to understand the transfer function of PID controller, motor and load. This model allows us to understand the connection between the zeros and the poles and the various parameters of the PID control. The simulations on the model reproduce the behavior of the control system during the tuning test. Considering that all the machines will have the same drive, it's possible to simulate different application changing the motion profile and motor and load parameters.

We need also model reproduces the interpolation that the control system does from the PLC up to the drive. This model generates a motion profile with a fine interpolation which is the input for the model in fig.7.

The final design of one complete side of the drive unit including the mechanism is shown in fig.8.

V. Conclusions And Future Developments

We have presented a model based systems engineering approach to the design of engineering systems that is particularly suited for mechatronic systems since it quite naturally addresses the complexity related to the concurrent, synergetic interactions of different engineering fields and to their control. Main features of the approach are the detailed analysis of the process to be implemented by the system under development and the central role played by the functional decomposition of such process. Furthermore the proposed model is strictly requirement-driven and fully supported by modeling and simulation. This enables clear traceability of design decisions to corresponding development needs and defines a clear validation and verification path.

Despite its logical and simple structure, however this proposal shows some strong limitations at the present stage. First of all we are presently forced to use a multiplicity of models built using very different technologies (multibody systems, block diagrams, finite elements), and very different modeling perspectives, with many difficulties to harmonize results and transfer information from one modeling environment to the other. Furthermore in practice to combine these models for co-simulation approaches is very limited and difficult. The question of multidisciplinary model building and configuring according to different levels of maturity of the design process has already been addressed in literature, see for example [15] and [16]. However, apart from the academic and intellectual value of these proposals and their great potential, there is still very little practical experience in industrial implementation and use. Furthermore we feel the need to follow and support the requirements engineering process, system development activities and their relations with simulation models and results using a more structured language. Finally we need to include into our design process software development that today follows a separate stream based on the use of UML [17] as modeling language.

For these reasons we are now starting the evaluation of some abstract but more general modeling approach which could enable us to organize the various models we use in a hierarchy, and at the same time provide us with a simple way to map the requirements that were originated from the aforementioned functional analysis onto the specific architecture of the system under design. For this purpose, we are investigating the possibility to introduce SysML [20] in order to manage this high-level description of the system [21] and at the same time keep the semantics consistent with the efforts done in software development. Another line of research involves the introduction of a single language to describe the low-dimensional model, i.e. a lumped-parameter representation of the system to be developed: a promising choice seems to be the use of bond graphs [22], which allows the construction of multiphysics models in a pictorial and a-causal way; first results in this line of development are presented in [18]. It is reasonable to assume that the success of both these attempts will be greatly influenced by the consensus they will be able to gather among potential users and by the number and quality of software tools that have been or will be developed to aid the formal verification of design and modeling choices.

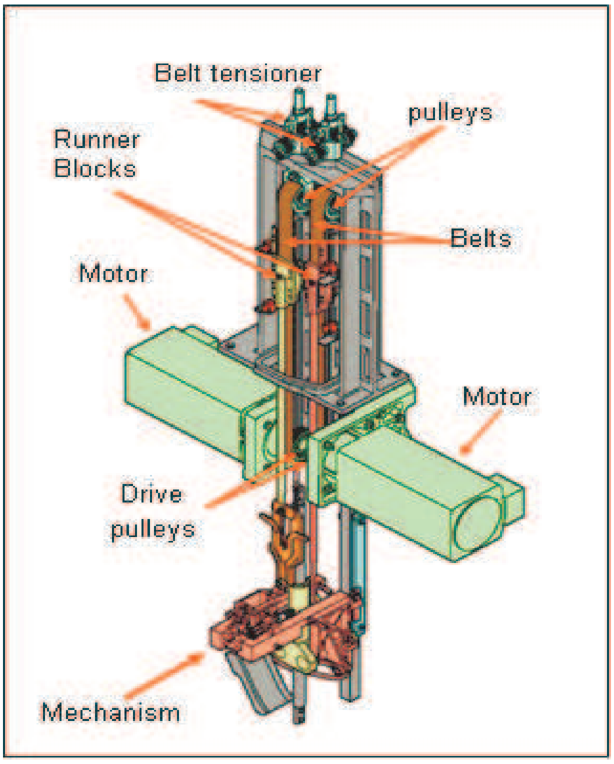

Fig. 8 Complete Drive Unit (one side)

ACKNOWLEDGMENT

The authors would like to thank Dr. A. Vollerthun e M. De la Cruz, 3D Systems Engineering GmbH, Munich, for their support in the development of the functional model.

REFERENCES

[1] N. P. Suh, *The Principles of Design*, Oxford University Press, 1990.

[2] D. Tate and M. Nordlund, *A Design Process Roadmap as General Tool for Structuring and Supporting Design Activities*, Proceeding of the Second World Conference on Integrated Design and Process Technology, Soc. for Design and Process Science, Austin, Texas, Dec. 1-4, 1999, pp. 97-104.

[3] M. M. Andreasen, The *Theory of Domains*, Technical Report, Institute for Engineering Design, Technical University of Denmark, 1992.

[4] S. Pugh, *Total Design*, Addison-Wesley, 1991.

[5] G. Pahl and W. Beitz, *Engineering Design*, Springer Verlag, 1996.

[6] D. M. Buede, *The Engineering Design of Systems*, Wiley & Sons, 2000.

[7] VDI 2206, *Design Methodology for mechatronic systems – VDI 2206*, Beuth Verlag GmbH, Verein Deutscher Ingenieure, Dusseldorf, 2004.

[8] R. Borsari, A. Vollerthun and M. de la Cruz, *12 Ways to Use a Functional Model*, paper submitted to EuSEC, European Systems Engineering Conference 2006, 18-20 September 2006, Edinburgh, UK.

[9] Akiyama, K., *Function Analysis - Systematic Improvement of Quality and Performance*, Cambridge, MA, Productivity Press, 1991.

[10] Gaso, B., *Modular Product Architectures*, Masters Thesis/Working Paper, Technische Universität München, Massachusetts Institute of Technology, 2001.

[11] Olson, D. G., Erdman, A. G., and Riley, D. R., *Mechanism and Machine Theory*, vol.20, n.4, 285-295, 1985.

[12] Borsari, R., Dunge, F., *Functional Design and Kinematic Synthesis of a Chain Driven High-Speed Packaging Machine*, 15th European ADAMS Users' Conference, Rome, 2000.

[13] Sacchetti, E., *Dynamical model of a mechatronic belt drive*, Diploma work, University of Modena, 2003 (in Italian).

[14] Sacchetti, E., *Motion control modeling and servo sizing: the design of mechatronic packaging machines*, Master Thesis, University of Modena, 2005.

[15] Diaz-Calderon A., *A Composable Simulation Environment to Support the Design of Mechatronic Systems*, Ph.D. Dissertation, Department of Electrical and Computer Engineering, Carnegie Mellon University, 2000.

[16] Andersson, K. and Sellgren, U., *MOSAIC (Integrated Modelling and simulation of physical behaviour of complex systems)*, NordDesign ´98, Stockholm, August 1998.

[17] Booch, G., Rumbaugh, J. and Jacobson, I., *The UML User Guide*, Addison- Wesley, 2000.

[18] Zanichelli, D., Secchi, C., Rubini, R., Fantuzzi, C., Bonfe', M., Borghi, D., Sacchetti, E. And Borsari, R., *Towards object oriented modeling of complex mechatronic systems for the manufacturing industry*, paper accepted, IMECE 2006, ASME 2006 International Mechanical Engineering Congress & exposition, November 5-10, Chicago, IL, USA.

[19] Secchi, C., Bonfé, M. and Fantuzzi, C., *On the use of UML for modeling mechatronic systems*, Transactions on Automation Science and Engineering, 2006, in press.

[20] *The SysML Specification*, version 1.0a, November 2005, http://www.sysml.org.

[21] Bassi, L., Secchi, C., Fantuzzi, C. And Bonfé. M., *An object-oriented approach to manufacturing systems modeling*, paper accepted at IEEE CASE 2006, October 7-10, Shanghai, China.

[22] Karnopp, D., Margolis, D. L. and Rosenberg, R. C., *System Dynamics*, John Wiley and Sons, 2000.