Proceedings of the XII International Symposium on Dynamic Problems of Mechanics (DINAME 2007),
Varoto, P. S. and Trindade, M. A. (Editors), ABCM, Ilhabela, SP, Brazil, February 26 - March 2, 2007

A Mechatronic Approach to Control of Stewart-Gough Parallel Manipulator

João Mauricio Rosario [1], Marcos Antonio Porta Saramago [1], Edgard de Oliveira [1], and Didier Dumur [2]

[1] UNICAMP – University of Campinas, 13083-970 – Campinas, SP, Brazil – rosario@fem.unicamp.br

[2] SUPELEC – Department of Automatic Control, 91190 – Gif-sur-Yvette, France – didier.dumur@supelec.fr

Abstract: This paper shows that a hierarchical architecture, distributing the several control actions in growing levels of complexity and using resources of reconfigurable computing, enables to take into account the easiness of future modifications, updates and improvements in robotic applications. An experimental device of a Stewart-Gough platform control, platform applied as a solution of countless practical problems, is presented using reconfigurable computing. The developed software and hardware are structured in independent blocks. This open architecture implementation allows an easy expansion of the system and a better adaptation of the platform to its related task.

Keywords: Stewart-Gough Platform, Mechatronics, Dynamics and Kinematics, Control, Robotics.

NOMENCLATURE

u = the control signal applied to the system,
y = the output of the system,
$\Delta(q^{-1}) = 1 - q^{-1}$ the difference operator,
A = polynomials in the backward shift operator q^{-1}, order n_a
B = polynomials in the backward shift operator q^{-1}, order of n_b
N_1 = the minimum prediction horizon,
N_2 = the maximum prediction horizon,
N_u = the control horizon,
W = transfer function factor

Greek Symbols

ξ = uncorrelated zero-mean white noise.
λ = the control weighting factor.

Subscripts

a relative to air

INTRODUCTION

The Stewart-Gough platform corresponds to a classical design for positioning and motion control, originally proposed in 1965 as a flight simulator, and still commonly used for that purpose. It is a parallel mechanism applied in a large variety of industrial problems like manufacturing of complex forms, aerospace, automotive, nautical, and machine-tool technology (Ollero, 2005). Researchers have examined many variants of the Stewart platform; most of them have six linearly actuated legs with varying combinations of leg-platform connections. Among many types of motion control platforms, this one appears of most interest being a widely accepted design for a motion control device.

Usually, six legs are spaced around the top plate and share the load on the top plate. This differs from serial designs, such as robot arms, where the load is supported over a long moment arm. The position and orientation of the mobile platform varies depending on the lengths to which the six legs are adjusted. This device can be used to position the platform in six degrees of freedom (three rotational and three translational degrees of freedom). In general, the top plate is triangularly shaped and is rotated 60 degrees from the bottom plate, allowing all legs to be equidistant from one another and each leg to move independently of the others.

This article presents a Stewart platform implementation based on reconfigurable computing applied to a Stewart-Gough platform implemented at the Automation and Robotics Laboratory, UNICAMP, Brazil. This system is used to simulate the movement of a sea tanker and within studies of cooperative robots.

Mathematical Description

Kinematics

The Stewart-Gough platform can accomplish a large number of complex tasks (Lee et al., 2003). It is a 6-degree of freedom parallel mechanism that consists of a rigid body top plate or mobile plate, connected to a fixed base plate through six independent kinematics legs. These legs are identical kinematics chains, composed of a universal joint, a linear electrical actuator, and a spherical joint. Typically, the legs are designed with an upper and lower body that can be adjusted, so that each leg has a variable length. The geometrical model of a platform expresses the position (X, Y, Z) and orientation (ψ, θ, ϕ) with respect to a fixed coordinate system linked at the base of the platform (Fig. 1), as function of its generalized coordinates (joints linear movements), that is:

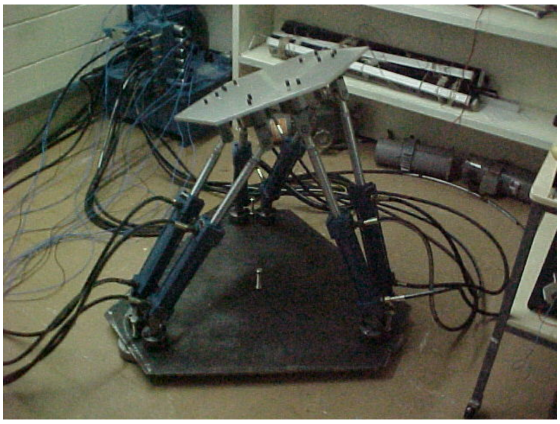
$$X_i = f(L_i) \tag{1}$$

where $L_i = (L_1 \;\; L_2 \;\; \cdots \;\; L_6)$ are the joints linear position, $X_i = (X \;\; Y \;\; Z \;\; \psi \;\; \theta \;\; \phi)$ the position-orientation vector of a point of the platform, and

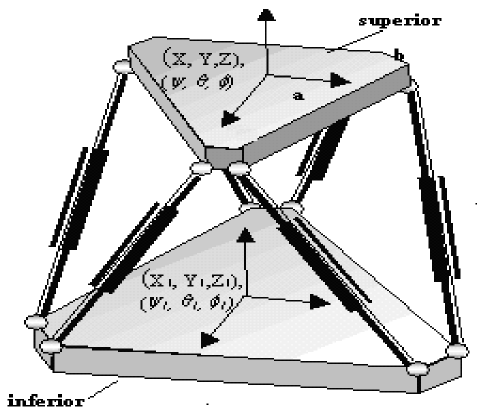
$$\begin{aligned} T(\psi,\theta,\phi) &= \mathrm{rot}(x,\phi)\,\mathrm{rot}(y,\theta)\,\mathrm{rot}(z,\psi) \\ &= \left[\begin{array}{c|c|c} c\phi\, c\theta & -c\phi s\theta s\psi - s\phi\, c\psi & c\phi\, s\theta\, c\psi + s\phi\, s\psi \\ s\phi\, c\theta & -s\phi s\theta\, s\psi + c\phi\, c\psi & s\phi\, s\theta\, c\psi - c\phi\, s\psi \\ -s\theta & c\theta s\psi & c\theta c\psi \end{array}\right] \end{aligned} \tag{2}$$

$$\theta = ATAN2\left[\frac{-n_Z}{c\phi\, n_x + s\phi\, n_y}\right], \; \phi = ATAN2\left[\frac{n_y}{n_x}\right], \; \psi = ATAN2\left[\frac{s\varphi\, a_x - c\varphi\, a_y}{-s\varphi\, s_x + c\varphi\, s_y}\right]$$

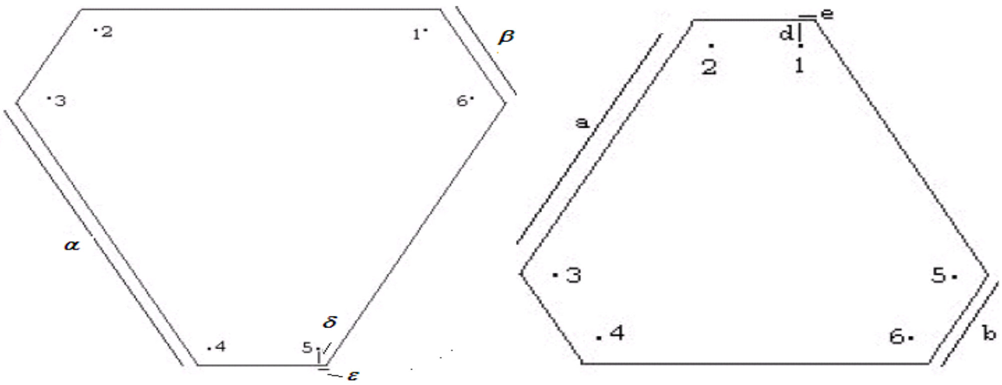
$\mathrm{n} = \lfloor n_x \;\; n_y \;\; n_z \rfloor$, $\mathrm{s} = \lfloor s_x \;\; s_y \;\; s_z \rfloor$, $\mathrm{a} = \lfloor a_x \;\; a_y \;\; a_z \rfloor$: orthonormal vector that describes the orientation.



a) Experimental Implementation.



b) Referential Coordinate system.



Inferior base ***Superior base***

c) Actuators reference points.

Figure 1: Stewart Platform.

This transformation matrix (T) can be interpreted as the one that transforms the vector associated with each linear actuator into a new configuration, with the addition of a corresponding term related to the translation movement (Rosario, et al., (2006a). To derive the kinematic model, the superior part of the base has been idealized as an irregular hexagon, each vertex of this hexagon corresponding to an actuator connection, as shown in Fig. 1. Further, the points that determine the movement of the superior base are the extremities of the six linear actuators settled in the inferior base of the platform. Therefore, assuming that the linear actuators have reached a final position and orientation, the problem consists in calculating the center of mass coordinates of the superior base and the RPY angles of orientation (roll, pitch and yaw), in relation to this reference system.

The relative positions of each point of attachment of the linear actuators can be derived from the parameters of movement and orientation, leading to new positions for the superior extremities of the linear actuators through an analytical calculation procedure.

The position vector of the linear actuator for the upper/lower base, $P_i,\ P_s$, determined in relation to the reference system fixed at the centre of mass of the inferior part, is described through equation 3. The parameters $\alpha,\beta,\delta,\varepsilon,a,b,d,e$ are reported in Fig. 2, h represents the position of the centre of mass of the superior base in the initial configuration, and each line of $P_i,\ P_s$ represents the coordinates of the inferior ($A_1\ \cdots\ A_6$) and superior ($B_1\ \cdots\ B_6$) extremities of the actuators.

$$P_i = \begin{bmatrix} A_i+\varepsilon & D_i-\delta & 0 \\ -A_i+\varepsilon & D_i-\delta & 0 \\ -A_i+\varepsilon+C_i & D_i-\delta+C_i & 0 \\ -B_i+\varepsilon & D_i+\delta & 0 \\ B_i+\varepsilon & -D_i+\delta & 0 \\ A_i-\varepsilon-C_i & D_i-\delta+C_i & 0 \end{bmatrix} \qquad P_s = \begin{bmatrix} A_s+e & D_s-d & h \\ -A_s+e & D_s-d & h \\ -A_s+e+C_s & D_s-d+C_s & h \\ -B_s+e & D_s+d & h \\ B_s+e & -D_s+d & h \\ A_s-e-C_s & D_s-d+C_s & h \end{bmatrix} \tag{3}$$

with $A_i = \frac{1}{2}\alpha$, $A_s = \frac{1}{2}b$, $B_i = \frac{1}{2}\beta$, $B_s = \frac{1}{2}a$, $C_i = 2(\varepsilon - B_i)\cos(t)$, $C_s = 2(e - B_s)\cos(t)$, $D_i = (A_i + B_i)\cos(t)$, $D_s = (A_s + B_s)\cos(t)$

Each linear actuator is associated to a position vector $\underline{X_i}$ considering inferior extremity of the position vector for each actuator and the value of the distension associated with actuator *i*. With $T(\psi,\theta,\phi)$ the previous transformation matrix, $\underline{X_i^T}$ is the new associated position vector for each upper position *i*:

$$\underline{X_i} = T(\psi,\theta,\phi)\ \underline{X_i^T} \tag{4}$$

From the knowledge of the position of the superior base, the coordinates of the superior extremities of the linear actuators are determined by the procedures previously described, resulting in a new position, whose norm corresponds to the new size of the actuator. If $\underline{X_0}$ is the reference point, the difference between the current sizes and the desired ones is the distension that must be imposed to each actuator to reach the new position:

$$\Delta L = \left|\underline{X_i^T} - \underline{X_0}\right| - \left|\underline{X_i} - \underline{X_0}\right| \tag{5}$$

Thus, the distance of the inferior extremity of the linear actuator up to the superior extremity is calculated, where the same one is determined from the transformation of coordinates. The kinematic model of the platform needs to receive the translation information in the form of a vector and the rotation matrix in RPY angles. This model enables to determine the appropriate axes lengths for the linear actuators so that the platform acquires the desired positioning (*x*, *y* and *z* coordinates, variable $j = 1,\cdots,3$). Eqs. 6 and 7 describe respectively the length of each linear actuator *k* connected to the upper mobile base before and after movement:

$$L = \sqrt{\sum_{j=1}^{3} (P_s^{kj} - P_i^{kj})^2} \quad \text{with} \quad k = 1,\cdots,6 \tag{6}$$

$$L + \Delta L = \sqrt{\sum_{j=1}^{3} (T_j^{-1}(\psi,\theta,\phi) P_s^{kj} - P_i^{kj})^2}$$

Inverse Kinematics

The reference input is defined through a set of displacements associated to position/orientation of the centre of the platform. After interpolation, these displacements will act as reference signals for positioning controllers located at each joint, that compare the signals derived from the position sensors of the joints (Spong, 1999, Pimenta, 2001). The calculation of references in angular coordinates, referring to the tasks defined in the Cartesian space, is expressed mathematically by the inversion of the kinematic model, that is:

$$\Delta L_i = J^{-1}(\Delta \underline{X}) \tag{8}$$

The controller makes corrections based on the dynamic model of the studied platform. The control structure of the joints, including the kinematic model and the control algorithms, is presented on the block diagram of Fig. 2. The kinematic conditions may generate a system of nonlinear equations resulting in complex solutions (Karger, 2003, Bonev, 2003). Simplifications in the direct kinematics model are usually approached in the attempt of accomplishing the control of this category of manipulators. In this work, the inverse kinematics model is solved with Jacobian form with coupling the equations associated to each joint movement.

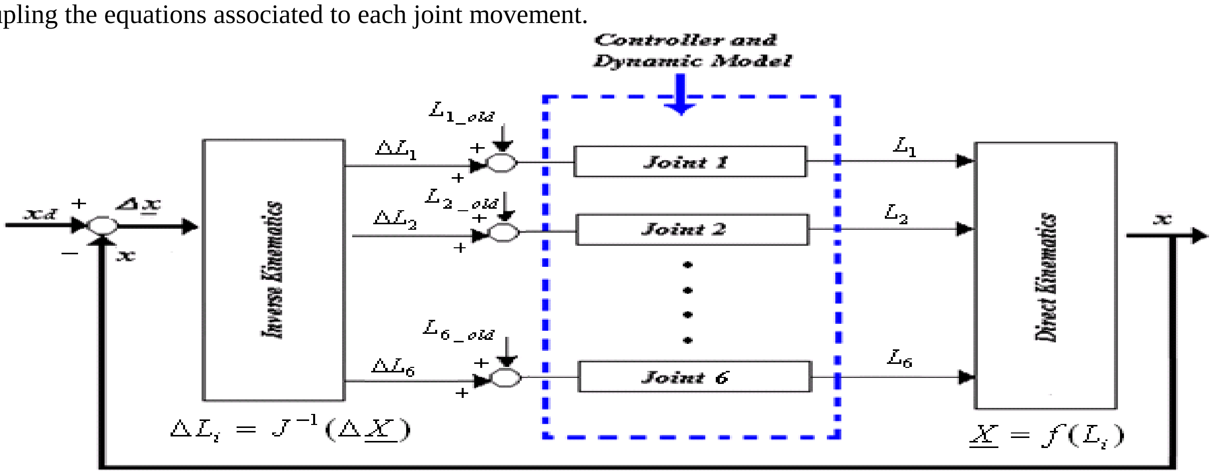


Figure 2: Kinematics Control Structure.

Dynamics

The control of movements can be accomplished by the composition of individual movements of each electrical actuator; the study of the dynamical and control systems is consequently realized for each joint. To take the coupling effects into account, and to solve the trajectory problem, the dynamic control involves the determination of the inputs, so that the drive of each joint moves its links to the position values with the required speed.

The dynamic model of a 6-DOF platform can be derived through the Euler-Lagrange formulation that expresses the generalized torque (David, et al., 1998). The dynamic model is described by a set of differential equations called dynamic equations of motion:

$$\tau_i = J_i \ddot{L}_i + F_i \dot{L}_i + \Gamma_i \qquad i = 1, \cdots, 6 \tag{9}$$

where $\tau_i(t)$ is the generalized torque vector, $L_i(t)$ the generalized frame vector (linear joints), $J_i(t)$ the inertial matrix, $F_i(t)$ the non-linear forces (for example centrifugal) matrix, Γ_i the gravity terms matrix.

Actuator Model

Each joint commonly includes a DC motor, a transmission system and an encoder. Considering the DC motor, the three classical equations are the following:

$$\begin{aligned} u(t) &= L_{mot}\frac{\mathrm{d}\, i(t)}{\mathrm{d}\, t} + R_{mot}\, i(t) + K_E \frac{\mathrm{d}\, \theta_m(t)}{\mathrm{d}\, t} \\ T_m(t) &= J_{eq}\frac{\mathrm{d}^2\, \theta_m(t)}{\mathrm{d}\, t^2} + B_{eq}\frac{\mathrm{d}\, \theta_m(t)}{\mathrm{d}\, t} \\ T_m(t) &= K_T\, i(t) \end{aligned} \tag{10}$$

where $T_m(t)$ is the torque, $\theta_m(t)$ the angular position of the motor axes, $i(t)$ the current, L_{mot}, R_{mot} respectively the inductance, resistance, J_{eq}, B_{eq} the inertia, friction of axis load calculated on the motor side.

A specific library has been elaborated, which includes complete axis models with controllers, motor drive, gearboxes and mechanical parts. This library enables easy change of controller's structure or motor specification.

Axis Control Structure

One advantage of the virtual environment that can be developed based on the previous model is the possibility of implementing and testing advanced axis control strategies, in particular Predictive Control, well known structure providing improved tracking performances. This philosophy, aiming at creating an anticipative effect using the explicit knowledge of the trajectory in the future, can be summarized as follows (Clarke, 1987, Boucher, 1995):

- Definition of a numerical model of the system, to predict the future system behaviour,
- Minimization of a quadratic cost function, over a finite future horizon, using future predicted errors,
- Elaboration of a sequence of future control values; only the first value is applied both on the system and on the model,
- Repetition of the whole procedure at the next sampling period according to the receding horizon strategy.

Model Definition

The CARIMA (Controlled AutoRegressive Integrated Moving Average Model) form is used as numerical model of the system, in order to cancel the steady state error, in case of a step input or disturbance, by introducing an integral term in the controller (Dumur et al., 1999). The predictive control law uses an external input-output representation form, given by the polynomial relation:

$$A(q^{-1})y(k) = B(q^{-1})u(k-1) + \frac{\xi(k)}{\Delta(q^{-1})} \tag{11}$$

where u is the control signal applied to the system, y the output of the system, $\Delta(q^{-1}) = 1 - q^{-1}$ the difference operator, *A*, *B* polynomials in the backward shift operator q^{-1}, of respective order n_a and n_b, ξ an uncorrelated zero-mean white noise.

Prediction Equation

The predictive methodology requires the definition of an optimal j-step ahead predictor which enables to anticipate the behavior of the process in the future over a finite horizon. From the input-output model Eq. 11, a polynomial predictor is designed under the following form:

Unknown polynomials F_j, G_j, H_j and J_j, corresponding to the expression of the past and of the future, are derived solving Diophantine equations, with unique solutions (Bonev, 2003).

$$\hat{y}(k+j) = \underbrace{F_j(q^{-1})y(k) + H_j(q^{-1})\Delta u(k-1)}_{\text{free response}} + \underbrace{G_j(q^{-1})\Delta u(k+j-1) + J_j(q^{-1})\xi(k+j)}_{\text{forced response}} \tag{12}$$

Cost Function

The GPC strategy minimizes a weighted sum of square predicted future errors and square control signal increments:

$$J = \sum_{j=N_1}^{N_2} \left(\hat{y}(k+j) - w(k+j)\right)^2 + \lambda \sum_{j=1}^{N_u} \Delta u(k+j-1)^2 \tag{13}$$

Assuming $\Delta u(t+j) = 0$ for $j \geq N_u$. Four tuning parameters are required: N_1, the minimum prediction horizon, N_2 the maximum prediction horizon, N_u the control horizon and λ the control weighting factor.

Cost Function Minimization

The optimal j-step ahead predictor Eq. 12 is rewritten in a matrix form:

$$\hat{\mathbf{y}} = \mathbf{G}\,\tilde{\mathbf{u}} + \mathbf{if}\,(q^{-1})\,y(t) + \mathbf{ih}(q^{-1})\,\Delta u(t-1) \tag{14}$$

with:

$$\mathbf{if}\,(q^{-1}) = \begin{bmatrix} F_{N_1}(q^{-1}) & \cdots & F_{N_2}(q^{-1}) \end{bmatrix}', \mathbf{ih}\,(q^{-1}) = \begin{bmatrix} H_{N_1}(q^{-1}) & \cdots & H_{N_2}(q^{-1}) \end{bmatrix}'$$

$$\tilde{\mathbf{u}} = \begin{bmatrix} \Delta u(t) & \cdots & \Delta u(t+N_u-1) \end{bmatrix}', \hat{\mathbf{y}} = \begin{bmatrix} \hat{y}(t+N_1) & \cdots & \hat{y}(t+N_2)) \end{bmatrix}'$$

$$\mathbf{G} = \begin{bmatrix} g_{N_1}^{N_1} & g_{N_1-1}^{N_1} & \cdots & \cdots \\ g_{N_1+1}^{N_1+1} & g_{N_1}^{N_1+1} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ g_{N_2}^{N_2} & g_{N_2-1}^{N_2} & \cdots & g_{N_2-N_u+1}^{N_2} \end{bmatrix}$$

The future control sequence is obtained minimizing the criterion Eq. 13 (Bonev, 2003):

$$\tilde{\mathbf{u}} = \mathbf{M}\left[\mathbf{w} - \mathbf{if}(q^{-1})\, y(t) - \mathbf{ih}(q^{-1})\, \Delta u(t-1)\right] \tag{15}$$

with: $\mathbf{M} = \mathbf{Q}\ \mathbf{G}'$, $N_u \times (N_2 - N_1 + 1)$ matrix,

$$\mathbf{Q} = \left[\mathbf{G}'\mathbf{G} + \lambda\, \mathbf{I}_{N_u}\right]^{-1} \text{ and } \mathbf{w} = \left[w(t+N_1) \quad \cdots \quad w(t+N_2)\right]'$$

Only the first value of the sequence Eq. 15 is finally applied to the system according to the receding horizon strategy.

RST Form of the Controller

The minimization of the previous cost function (Dumur, 1994) results in the predictive controller derived in the RST form according to Figure 3 and implemented through a difference equation:

$$S(q^{-1})\Delta(q^{-1})u(t) = -R(q^{-1})y(t) + T(q)w(t) \tag{16}$$

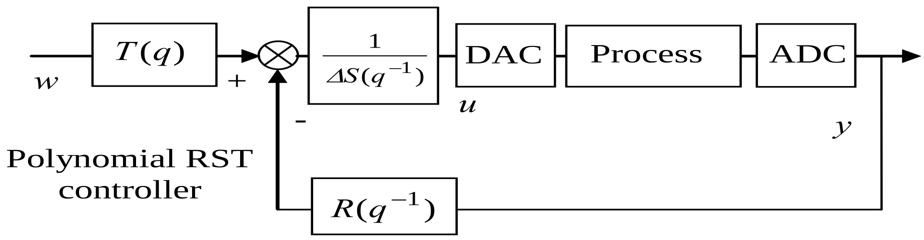


Figure 3: GPC in a RST form.

The main feature of this RST controller is the non causal form of the *T* polynomial, creating the anticipative effect of this control law. The degrees of the three polynomials are as follows:

$$\text{degree}\left(R(q^{-1})\right) = \text{degree}\left(R(q^{-1})\right)$$
$$\text{degree}\left(S(q^{-1})\right) = \text{degree}\left(B(q^{-1})\right)$$
$$\text{degree}\left(T(q)\right) = N_2$$

The GPC has shown to be an effective strategy in many fields of applications, with good time-domain and frequency properties (small overshoot, improved tracking accuracy and disturbance rejection ability, good stability and robustness margins), able to cope with important parameters variations.

Supervision and Control Architecture

Rapid Prototyping

The objective of this reconfigurable architecture concept is thus to enable an easy and quick adaptation and expansion of the system to these technological evolutions, for a better portability and interchangeability of the final system. Through the division of the structure in small functional blocks, with very specific dedicated interfaces, the modularization of the project becomes efficient.

Among all fields related to the complete achievement of an embedded project, hardware and software technologies have rapidly improved. That is particularly true for the evolution of motors, sensors, microprocessors, communication interfaces and power interfaces. From this, the idea is then to elaborate open structures, which may adapt very easily to the developments of all these technologies. The consequence of this requirement is the design of small independent modules, with communication interfaces, included within an open architecture oriented structure.

Using parameters of the above system, the global viability of the project has been assessed first through a dedicated virtual environment before experimental validation. However, the process of developing and implementing control strategies, including tuning phases, for this type of complex mechatronics system is extremely time-consuming (Cassemiro, et al., 2005).

In this direction, the rapid prototyping tools allow the design of integrated environments for modeling, simulating, and rapid prototyping algorithm development, with components that a) simulate the dynamic models of the complex mechatronics systems, b) perform a complex simulation of the overall mechatronics system and environment, c) automatically generate code for embedded robot control and d) communicate with the platform and control it remotely.

Supervision and Control

The proposed control architecture is a set of hardware and software modules, implemented with emphasis in rapid prototyping systems, integrated to give support to development of the platform tasks (Lima et al., 2000). The architecture is organized in several independent blocks, connected like a hierarchical structure in three control levels (Fig. 5):

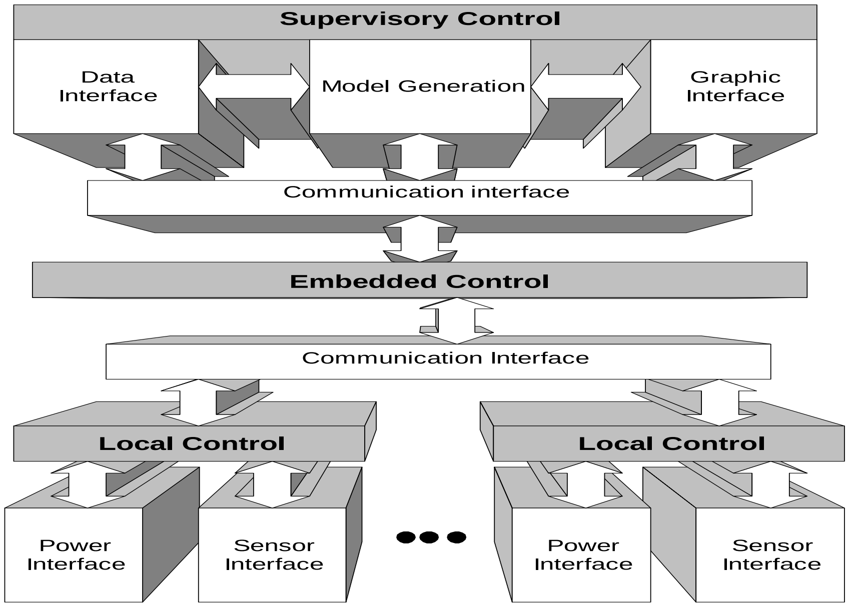


Figure 4: Stewart-Gough Platform – Control Architecture.

- **Supervisory control:** in this higher control level, the supervision of a generic platform task can be carried out, through the execution of global control strategies. This level also allows establishing corrections in the task realization according to the sensors data information.
- **Embedded control:** this level is dedicated to the embedded software for control. The control strategies allow decision making to be performed at a local level, with occasional corrections from the supervisory control level.
- **Local control:** this area is restricted to local control strategies associated with the sensors and actuators interfaces. The strategies in this level can be implemented under the rapid prototyping framework, through FPGA, as described below. The embedded controllers may be implemented under difference equations (RST form), which appear to be a very general and useful structure in an open architecture environment, including, for example, classical PID controllers as well as more advanced control techniques such as predictive control.

Simulation and Experimental Implementation

Position Controller using FPGA

An alternative to controllers implemented by software is the implementation using reconfigurable logic (Rosario, 2003b). The proposed controller has for objective the control of a platform with linear actuators. This programmable controller is able to process the digital signals originating from an encoder coupled to each linear actuator (ENCODER) and the digital signals of a trajectory (TRAJECTORY). For example, a PID digital controller written in a RST form can be implemented in PLD, with the gain parameters fitted through external programming. The controller’s output is a digital signal for the PWM power block.

The control of just one actuator is represented in Fig. 5, but the synchronized control of whole actuators can be easily achieved through the same PLD. Four main blocks are observed:

- **Error Detecting Block:** comparison of the signs ENCODER and TRAJETORY.
- **PID Controller Block:** PID digital controller, using the gain parameters incorporated in the control registers.
- **Control Register Block:** responsible for the parameters programming in PLD.
- **Power Interface Block:** conversion the binary word supplied by PID controller digital signals for further use by PWM power block.

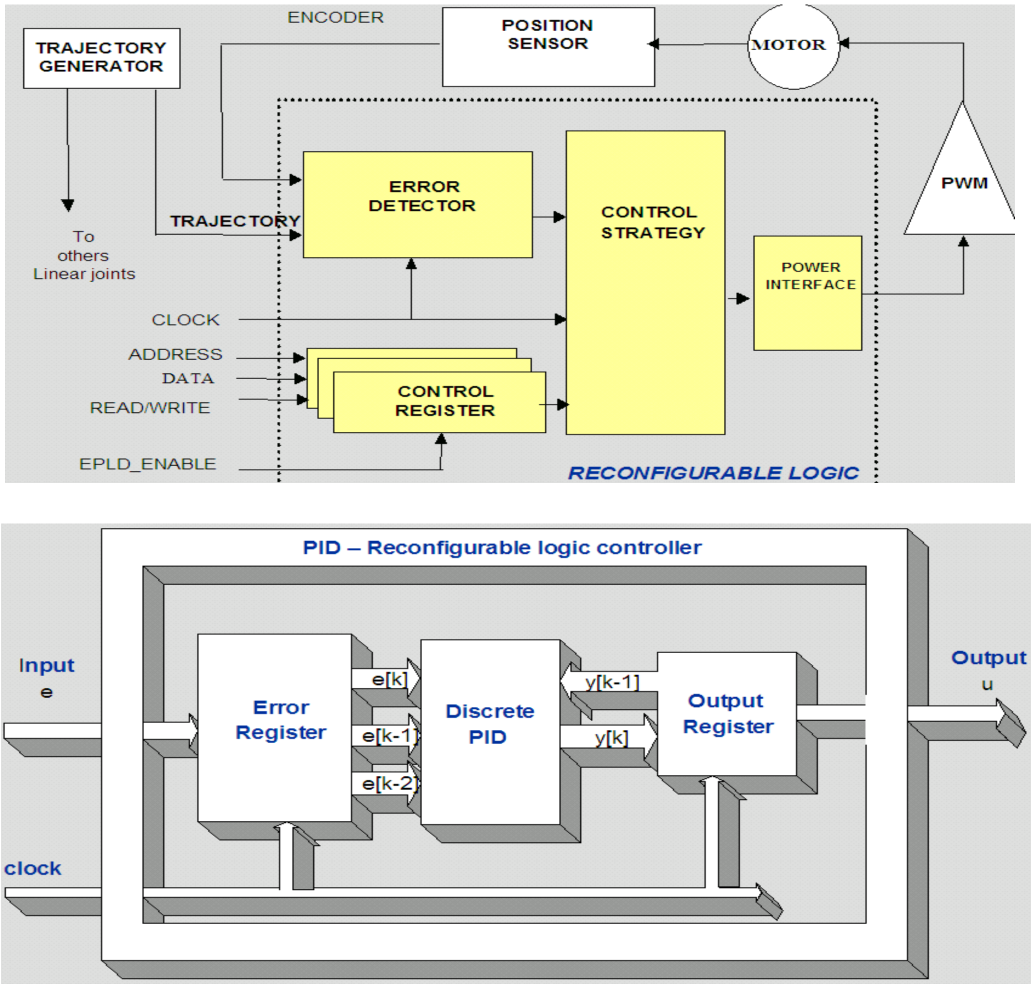


Figure 5: Control block diagram of the system and digital controller implemented in FPGA.

Prototyping Environment

A simulation scenario was developed for the environment related to the 6 DOF parallel manipulator, including motor drives, gear boxes, kinematic and dynamic models, and design of the control system for three axes. Simulations described below consider trajectories issued from the path generation module. The model was tested first in Matlab-Simulink^{TM} language and the final control hardware implementation was performed in visual programming using LabVIEW^{TM} software (Fig. 6). This last one is used for communication purposes between the program and the control hardware of the prototype.

Kinematics Model

The development of a numerical algorithm (David et al., 1998), which aims at finding the linear positions for a task defined with respect of the platform centre in the Cartesian Space, contains the solution of the inverse kinematics through the use of recursive numerical methods based on the calculation of the kinematics model and of the inverse Jacobian matrix of the manipulator. This algorithm has been validated through different simulations, assessing the behavior of the trajectory (joint coordinate). For this purpose the kinematics model of the platform was used, with six linear joints.

Fig. 7a shows the joints movements of each linear actuator and the translation displacement (45 degrees, approximately) of one point of the upper base of this platform obtained through the inverse kinematics model (Fig. 7b). Fig. 8 shows results of the proposed simulation, obtained with PID axis controllers implemented through FPGA, considering general sea movements and LABVIEW^{TM} experimental platform.

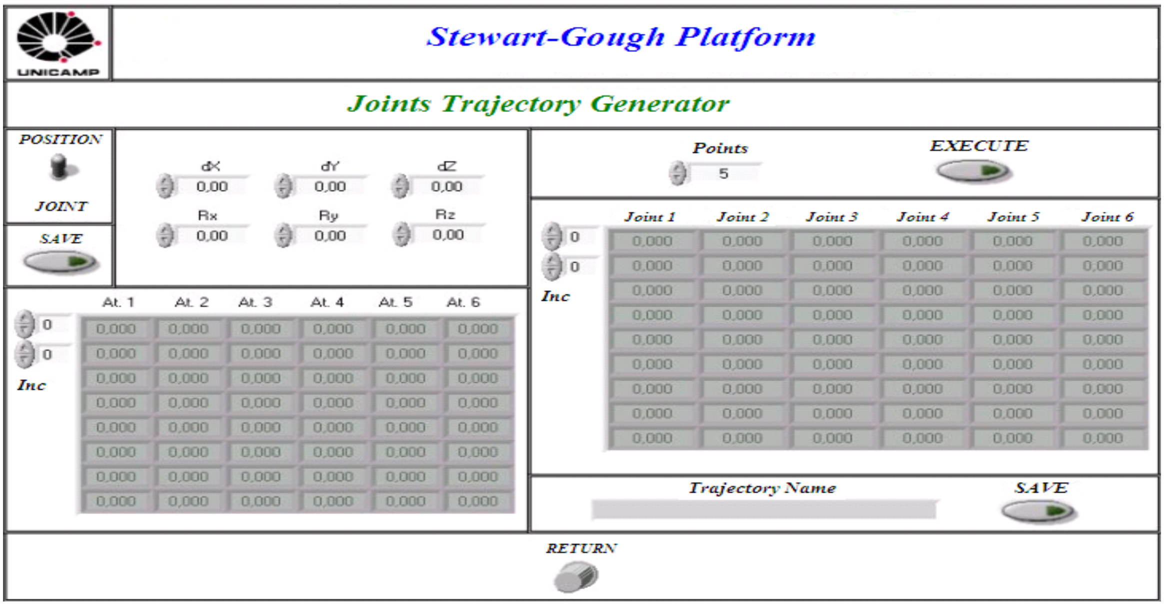
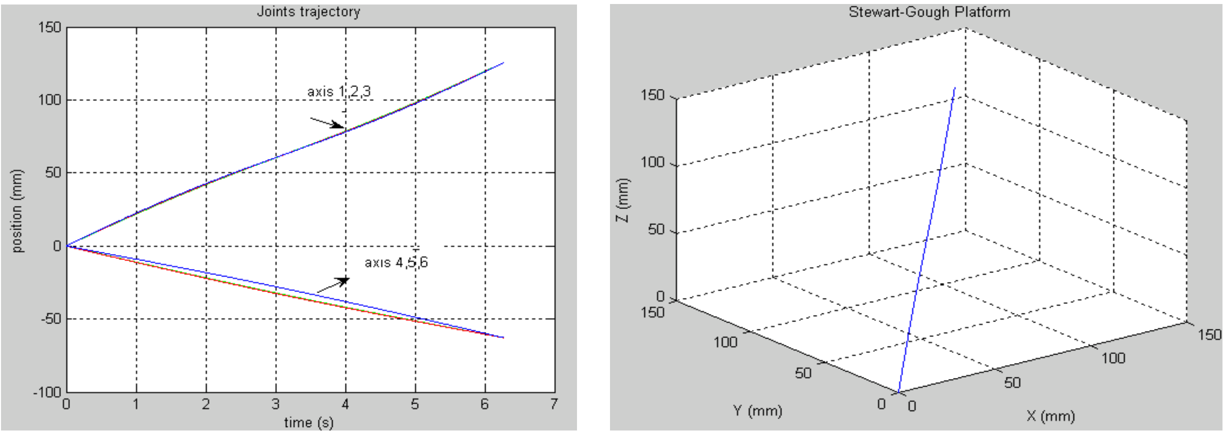


Figure 6: Kinematics model implemented in LabVIEW^{TM}.



a) Joints evolutions.

b) Trajectory description.

Figure 7: Kinematics model - Simulation results.

Conclusions

This paper presents the study of kinematics, dynamics and control of a Stewart-Gough platform, under a reconfigurable architecture concept, considering the division of the system in small functional blocks. This implementation consisted in merging knowledge acquired in multiple areas, and appears as a very promising design strategy for a better reconfigurability and portability of systems.

This platform also becomes a powerful benchmark for many research activities, such as the validation of controllers and supervision strategies, model generation and data transmission protocols, among others. For example, the implementation of predictive controllers on this prototype may enable the test of this advanced control strategy under severe conditions of use.

To simplify tests, implementation and future modifications, the use of rapid prototyping functions in the implementation of the interfaces and other logical blocks is emphasized in the proposed prototype. The control block, for example, can benefit of the characteristics of low consumption, high-speed operations, integration capacity, flexibility and simple programming. Some promising aspects of this architecture are:

- Flexibility, as there is a large variety of possible configurations in the implementation of solutions for several problems. It is a powerful tool for prototype design, allowing simple solution to control the several sensors and actuators usually present in this kind of projects,
- Open architecture of this platform enables the use for educational and researches activities, with possibility of modification of control strategies during operation of the platform.

ACKNOWLEDGMENTS

The authors acknowledge support of the CNPq and CAPES – Brazilian Council of Research, and ARCUS program, France, through a scientific collaborative work between the Control Department of SUPELEC, France and the Laboratory of Automation and Robotics, UNICAMP, Brazil.

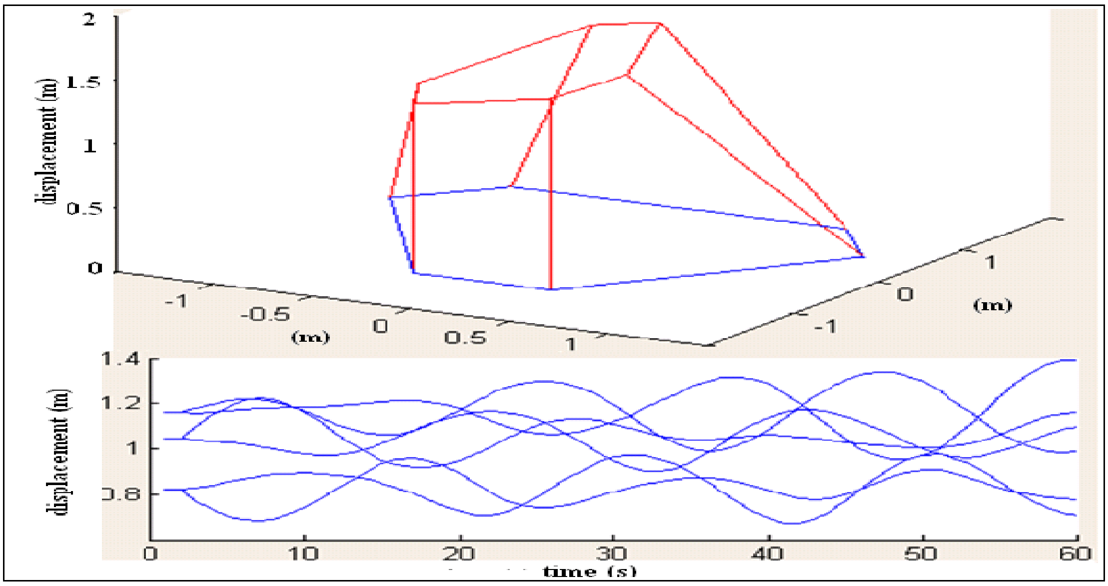


Figure 8: Joints outputs and corresponding spatial platform movements.

REFERENCES

A. Ollero, S. Boverie, R. Goodal, 2005, “Mechatronics, Robotics and Components for Automation and Control”, Annual Reviews in Control IFAC Journal, No 26, pp 203-228.

T.Y. Lee, J.K. Shim, 2003, “Improved analytic elimination algorithm for the forward kinematics of the general Stewart–Gough platform”, Mechanism and Machine Theory, vol. 38, pp. 563–577.

J.M. Rosario, E. Oliveira, D. Dumur, 2006, “Conception of Stewart-Gough Platform with Reconfigurable Control”, Proceedings of the International Symposium on Power Electronics, Electrical Drives, Automation and Motion – SPEEDAM 2006, pp. S10.13-S10.18.

M.W. Spong, M. Vidyasagar, 1989, “Robot Dynamics and Control”, John Wiley & Sons, New York.

K.B. Pimenta, J.P. Souza, J.M. Rosario, D. Dumur, 2001,“Control of Robotic Joints with Generalized Predictive Control (GPC)”, RADD’2001, Vienna, Austria.

A. Karger, 2003, “Architecture singular planar parallel manipulators”, Mechanism and Machine Theory, v.38, pages 1149-1164.

I.A. Bonev, 2003, “A closed-form solution to the direct kinematics of nearly general parallel manipulators with optimally located three linear extra sensors”, IEEE Trans. on Robotics and Automation, Vol. 17, No 2, pp. 148-156.

S. David, J.M. Rosario, 1998, “Modeling, Simulation and Control of Flexible Robots”, CONTROLO’98, Coimbra, Portugal , pp. 532-539.

D. W. Clarke, C. Mohtadi, P.S. Tuffs, 1987 “Generalized Predictive Control”, Part I "The Basic Algorithm", Part II "Extensions and Interpretation, Automatica, 23(2):137-160.

P. Boucher, D. Dumur, 1995, “Predictive Motion Control”, Journal of Systems Engineering, Special Issue on Motion Control Systems, vol. 5, Springer-Verlag, London, pp.148-162.

D. Dumur, P. Boucher, 1994, “New Predictive Techniques - Control Axis Solutions, Proceedings of the 3rd IEEE Conference on Control Applications”, vol. 3, pp. 1663-1668.

E.R. Cassemiro, J.M. Rosário, D. Dumur, 2005,“Robot Axis Dynamics Control using a Virtual Robotics Environment“, Preprints of IEEE Transactions on Systems, Man and Cybernetics - ETFA’05, Catania, Italy pp. 305-311.

C.R.E Lima, N.C. Silva, J.M. Rosário, 2000, “A Proposal of Flexible Architecture for Mobile Robotics”, 7th Forum International Conference and Mechatronics Education Workshop, Atlanta.

J.M. Rosário, J.M., C.R. Erig Lima, C.R., H. Ferasolli, R. Pegoraro, 2003, “Reconfigurable Architecture Proposal to Application on Mobile Embedded Systems Prototypes”, 7th IFAC Symposium of Robot Control – SYROCO´03, vol. I, 2003, pp. 89-94.

RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.