Bordbar, Behzad, Giacomini, Luisa and Holding, David J. (2000) *Hybrid Control Systems: a Design Case Study.* In: Proceedings of the 2000. IEEE International Conference on Control Applications. Conference Proceedings. IEEE, pp. 947-952. ISBN 0-7803-6562-3.

Hybrid Control Systems: a Design Case Study

B. Bordbar, L. Giacomini, D.J. Holding*
Aston University, Electronic Engineering,
Aston Triangle,
B4 7ET, United Kingdom
{b.bordbar, l.giacomini, d.j.holding}@aston.ac.uk

Abstract

The paper describes a design case study which explores the hybrid control of a distributed system comprising linked inverted pendulums. First a continuous controller is designed for each pendulum mechanism to provide low-level stabilization and profiled motion control. The paper then addresses the problem of developing a production-line style system using two loosely interconnected pendulum mechanisms and associated product-transfer manipulators. A supervisory system is developed using compositional methods and is modelled and analysed using controlled Petri nets. It is shown that using an appropriate coordination strategy it is possible to achieve a stability envelope for the composite system which is greater than that of the individual components. The function and performance of the system are demonstrated by simulation.

1. Introduction

The control of sets of independently driven mechanisms, from a simple conveyor belt to a sophisticated robotic manipulator, is traditionally dealt with in a continuous time/continuous state environment. When these mechanisms are used in a real environment, such as a production line, they are required to perform sequences of tasks and they may have to be synchronized with each other either to carry out joint tasks or to avoid unwanted interaction between the mechanisms. The control engineer conventionally copes with these requirements by embedding tests (which switch between regulators or reference trajectories) inside the continuous controller. Verification of the functionality of the overall controller is often left to extensive simulation.

In recent years, much research has been carried out on the concepts of *hybrid systems* and *hybrid control* [3], with the objective of developing an integrated approach to the discrete event and continuous parts of such systems. In many cases the need is to combine the study of the continuous domain stability/controllability with the study of the intended discrete domain functionality, such as the supervisory control of tasks sequences.

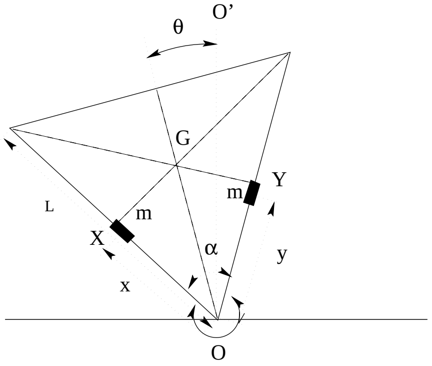
In this paper we report the use of discrete event tools combined with simple regulators, to design a hybrid controller for a distributed system with complex and reschedulable tasks sequences. The approach is demonstrated using a simple distributed system comprising two loosely-coupled independently-driven mechanisms (or components) which take the form of a pair of coupled inverted pendulums. We start by analysing the static and dynamic behaviour of each of these components in isolation and then design individual controllers for set-point regulation. An important part of the design procedure is the process of abstracting conventional switching decisions from the continuous system and making them an explicit part of the discrete event layer, so that they can be modelled and analysed using discrete event methods such as Automata or Petri nets [2, 5, 6].

The paper describes the development of a supervisory control system for the two loosely interconnected pendulum mechanisms and associated product-transfer manipulators using Petri net techniques. The Petri net model of the composite system was then analysed using Petri net techniques to verify the behaviour of the system. Finally, the analysis is extended to examine the effect of severe disturbances, such as those which might cause a pendulum to move into an unstable state (falling or toppling) and which, unless controlled, might cause a 'domino-effect' instability in the second component. It is shown that using an appropriate decision and coordination strategy it is possible to achieve a stability envelope for the composite system which is greater than that of the individual components. Moreover, the reachability graph of the Petri net can be backtracked to identify behaviour (and associated physical parameters) that leads to hazards such as livelock (caused by a deadly mutual embrace of the two inverted pendulums).

2. Introduction to the design problem: an inverted pendulum

Consider an inverted pendulum formed by a rotating triangular frame and two balance weights, as shown in Fig. 1. The triangle is assumed isosceles and to lie in the vertical plane. The triangular frame can rotate freely about an axis through the origin O. The mass of the frame is considered to be concentrated in the centre of mass of

*Corresponding Author



x distance of mass X from vertex O

y distance of mass Y from vertex O

α angle between the two rails (fixed)

θ angle between OG and OO'

L length of the rails (fixed)

Figure 1: The triangular frame

the triangle. The two balance masses X and Y can move along the sides of the frame: we will call these two sides 'rails'. The movement of X and Y along the rails changes their centre of mass and thus the balance of the triangular frame system. The triangle system is in equilibrium if it is at rest and the effective centre of mass of the frame, X and Y lies on the vertical line OO' (Fig. 1).

2.1. Dynamics of a single inverted pendulum

Consider the kinematic and dynamic equations of an inverted pendulum of the form described above. Applying the Euler-Lagrange procedure, and taking into consideration some simplyfing assumptions such as the absence of friction and damping terms, the following equations have been derived:

$$\ddot{x} = \frac{F_x}{m} + x\dot{\theta}^2 - gcos\left(\theta - \frac{\alpha}{2}\right), \tag{1}$$

$$\ddot{y} = \frac{F_y}{m} + y\dot{\theta}^2 - gcos\left(\theta + \frac{\alpha}{2}\right), \tag{2}$$

$$\begin{aligned}\ddot{\theta} &= \frac{1}{\frac{J}{m} + x^2 + y^2}\Big[- \dot{x}\dot{\theta}x - \dot{y}\dot{\theta}y + gxsin\left(\theta - \frac{\alpha}{2}\right) \\ &+ gysin\left(\theta + \frac{\alpha}{2}\right) + \frac{2ML}{3m}gcos\frac{\alpha}{2}sin\theta\Big], \qquad (3)\end{aligned}$$

where J is the inertia of the frame, F_x and F_y the forces applied to X and Y, respectively.

2.2. Static Analysis

If $x = y$, the system has two equilibrium points, one stable ($\theta = \pi$) and one unstable ($\theta = 0$). In the more general case, $x \neq y$, the frame will have an unstable equilibrium point at $\theta^* \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ and a stable one at $\pi - \theta^*$. Due to the geometry of the device, θ^* will belong to a finite sector of the plane.

Moving each of the two masses to the limit on the rails, i.e. $(x, y) = (0, L)$ and $(x, y) = (L, 0)$, we are able to determine the boundary values for θ^*. Specifically, if the system has initial conditions $(x(0), y(0), \dot{x}(0), \dot{y}(0), \theta(0), \dot{\theta}(0)) = (0, L, 0, 0, \theta_m, 0)$ or, equivalently, $(x(0), y(0), \dot{x}(0), \dot{y}(0), \theta(0), \dot{\theta}(0)) = (L, 0, 0, 0, \theta_M, 0)$, then, under the hypothesis that there are no disturbances, the limiting values for the equilibrium condition are:

$$\theta_M = -\theta_m = arctg\left(tan\frac{\alpha}{2}\left[\frac{1}{1 + \frac{2M}{3m}}\right]\right) \tag{4}$$

It follows that $(-\pi - \theta_m, \theta_m) \bigcup (\theta_M, \pi - \theta_M)$ is the smallest sector in the plane where the frame cannot be stabilized/controlled.

3. Dynamic stabilization and motion control

The primary control objectives are:

1. To make $\theta = \theta^*$ a stable equilibrium point for any $\theta^* \in [\theta_m, \theta_M]$.
2. To move the frame from the equilibrium angle $\theta = 0$ to the equilibrium angle $\theta = \theta^*$, by the use of X, Y movements.

Moreover, the controller should be robust towards external disturbances.

A controller has been designed to stabilize the frame in $\Omega \subset \{\theta \in \Re \mid \theta_m < \theta < \theta_M\}$, where the amplitude of the sector Ω depends on the performances of the regulator with respect to the initial condition for x, $\dot{x}$, y, $\dot{y}$, θ, $\dot{\theta}$. Consider the properties of system defined by Equations (1)–(3), where F_x and F_y are the forces generated by the actuators to move the masses up and down the rails, i.e. they are the controller outputs. Let the scheme in Fig. 2 be applied; r_1 and r_2 are the reference trajectories (control inputs) for the regulated rails, whilst θ_d is the reference trajectory for the variable θ.

In choosing an appropriate control strategy, it is of note that if the frame has to balance at an angle at the left of the line OO', the mass Y has to be moved up, and mass X has to be moved down. Alternatively, if θ^* is at the right of OO', mass X should be up, mass Y should be down. Therefore, it is sensible to generally link the two control inputs F_x and F_y in such a way that, when one says "up" the other one says "down". In practice, the reference inputs for the two masses were set as follows: $r_1 = \frac{L}{2} + r$ and $r_2 = \frac{L}{2} - r$ where r is a function of $e = \theta - \theta^*$, $\theta^* \in \Omega$, and its first derivative.

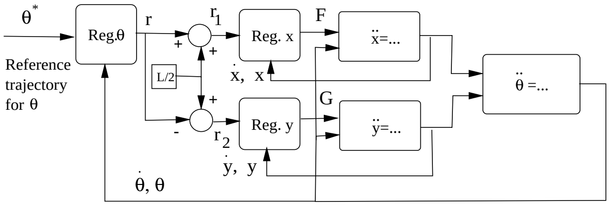


Figure 2: Control scheme

With reference to Fig. 2,

$$\begin{array}{ll} Reg.x & F_x = -a(x - r_1) - b\dot{x}\ , \\ Reg.y & F_y = -a(y - r_2) - b\dot{y}\ , \\ Reg.\theta & (r_1, r_2) = f(\theta, \dot{\theta})\ . \end{array}$$

As an example, for $m = M = 1\ kg$, $L = 1\ m$, and $\alpha = \frac{\pi}{2}$, the parameters of the control laws have been chosen as $a = 1000$, $b = 500$, $k_1 = 10000$, and $k_2 = 0.01$.

3.1. The θ regulator

As a matter of fact, it will be a waste of control effort to try to control the frame when it is outside the stabilizability zone. Morevover, if X and Y are left uncontrolled at a position different from (0, 0), the inertial velocity of the frame will not be minimal during the 'falling down' movement and this fact will be an obstacle to the introduction of the 'rescue policy' of Section 6. With this in mind, *Reg.θ* can be designed as a two module regulator (the control is switched between them) as follows

$$f(\theta, \dot{\theta}) = \begin{cases} (0,\ 0) & \theta \in (-\infty, -\theta_c] \bigcup [\theta_c, \infty) \\ (\frac{L}{2} + u(\theta, \dot{\theta}, \theta_d, \dot{\theta}_d), \frac{L}{2} - u(\theta, \dot{\theta}, \theta_d, \dot{\theta}_d)) & \\ & \theta \in (-\theta_c, \theta_c) \end{cases}$$

where $u(\theta, \dot{\theta}, \theta_d, \dot{\theta}_d)$ is a suitable set-point control, and $(\theta_d, \dot{\theta}_d)$ is the set-point. What need to be established is θ_c that is the limit of controllability angle. From the arguments in Section 2.2., it can be said that surely no control can stabilize the system if $\theta > \theta_M$ or $\theta < -\theta_M$, then θ_c is at most θ_M. A narrower interval can be found depending on the control law. For example, with some control strategies, there can be found the limits for controllability in terms of $\theta(0)$ and $\dot{\theta}(0)$. With a 'smooth' control achieving a first–order–like behaviour of the closed loop system, the limit set-point can be θ_M. In addition, even though the system is not stabilizable, the control procedure that will be applied in the region $(-\theta_M, \theta_M)$ will surely help to reduce the falling down angular speed. Following this reasoning, it has been decided that the law that will be used is

$$f(\theta, \dot{\theta}) = \begin{cases} (0,\ 0) & \theta \in (-\infty, -\theta_M] \bigcup [\theta_M, \infty) \\ (\frac{L}{2} + u(\theta, \dot{\theta}, \theta_d, \dot{\theta}_d), \frac{L}{2} - u(\theta, \dot{\theta}, \theta_d, \dot{\theta}_d)) & \\ & \theta \in (-\theta_M, \theta_M) \end{cases}$$

even though $(-\theta_c, \theta_c) \subset (-\theta_M, \theta_M)$.

Now, $u(\cdot)$ must be specified. We want precise set-point regulation, possibly with velocity control as well, and robustness towards bounded disturbances.

For this task, a discontinuous control has been chosen because of the highly non linearity of the system and the fast dynamics it can provide for the closed loop system.

3.2. Sliding mode control

Given a dynamical system $\dot{x} = f(x) + g(x)u$, $x \in \Re^n$, $u \in \Re$, a typical *sliding mode control* [7] is set up in two steps

1. Design a switching function $S(x)$, such that for $S(x) = 0$ some control objective is satisfied.

2. Design a control law $u(x)$ discontinuous on $S(x) = 0$ such that the states of the system reach the *sliding manifold* $\{x \in \Re^n \mid S(x) = 0\}$ in a finite time.

For step 1, a suitable surface for our system is

$$S(\cdot) = \dot{\theta} + c(\theta - \theta_d)$$

where θ_d is the angle set-point. Because the control u appears only in the second derivative of S (in fact $\dot{S} = \ddot{\theta} + \ldots$, and $\ddot{\theta}$ does not contain u, as can be seen from 3, while $\ddot{S} = \dddot{\theta} + \ldots$ contains it thorough the expressions of $\ddot{x}$ and $\ddot{y}$), we apply a second order sliding mode algorithm[1] [1], that is

1. Let $y_1 = S$ and $y_2 = \dot{S}$;

2. write the auxiliary system

   $$\begin{cases} \dot{y}_1(t) & = y_2(t) \\ \dot{y}_2(t) & = \mathcal{F}[y(t), t] + \mathcal{G}[y(t), t]u(t) \end{cases} \tag{5}$$

   with $y(t)^T = [y_1(t)\ \ y_2(t)]$, $y_2(t)$ unmeasurable, $\mathcal{F}[y(t); u(t); t] = \frac{\partial}{\partial x}\left(\frac{\partial S}{\partial x}(f(x) + g(x))\right)(f(x)+g(x)u)$, and $\mathcal{G}[y(t); t] = \frac{\partial S}{\partial x} g$ uncertain functions

3. find suitable bounds F, Γ_m and Γ_M such that

   $$|\mathcal{F}[y(t); t]| < F \tag{6}$$

   $$0 < \Gamma_m \le \mathcal{G}[y(t); t] \le \Gamma_M \tag{7}$$

4. apply the control law

   $$\begin{array}{l} u(t) = -\alpha(t) U_M \mathrm{sign}\left\{y_1(t) - \frac{1}{2} y_1(t_{M_i})\right\} \\ \alpha(t) = \begin{cases} \alpha^* & if\ \ [y_1(t) - \frac{1}{2} y_1(t_{M_i})] \times \\ & [y_1(t_{M_i}) - y_1(t)] > 0 \\ 1 & if\ \ [y_1(t) - \frac{1}{2} y_1(t_{M_i})] \times \\ & [y_1(t_{M_i}) - y_1(t)] \le 0 \end{cases} \end{array} \tag{8}$$

   where U_M is the control amplitude to be suitably selected, t_{M_i} is such that $y_2(t_{Mi}) = 0$, and $y_1(t_{M_i})$ represents the last extremal value of the $y_1(t)$ function, i.e., the last local maximum, local minimum or horizontal flex point of $y_1(t)$.

[1] Another solution, would be to choose the surface $S = \ddot{\theta} + c_1\dot{\theta} + c_2\theta$ [7], however this would require us to measure the angular acceleration.

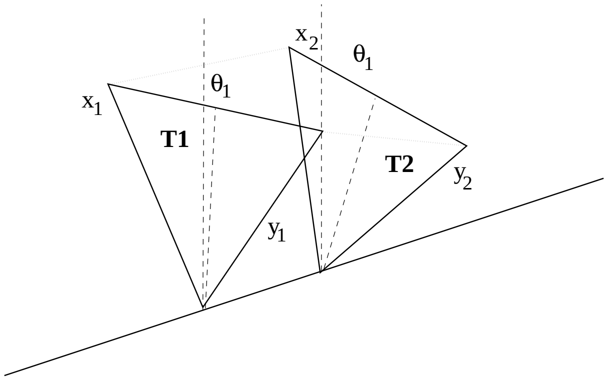


Figure 3:

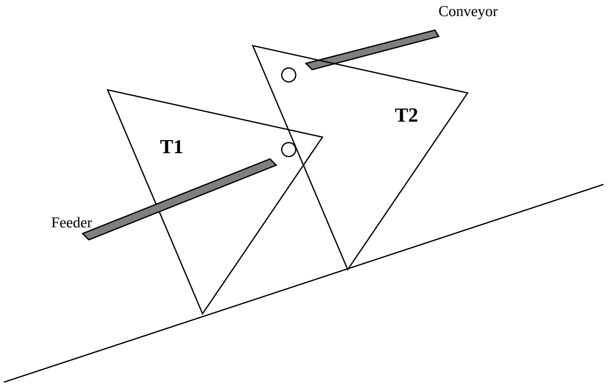


Figure 4:

In [1] it has been proved that, the corresponding sufficient conditions for the finite time convergence to the sliding manifold are

$$\left\{ \begin{array}{l} \alpha^* \in (0,1] \cap (0, \frac{3\Gamma_m}{\Gamma_M}) \\ U_M > \max\left(\frac{F}{\alpha^*\Gamma_m}, \frac{4F}{3\Gamma_m - \alpha^*\Gamma_M}\right) \end{array} \right. \tag{9}$$

4. Forming a composite system of two loosely coupled inverted pendulums

Consider a composite system formed by placing two inverted pendulum (of the triangular form described above) on the same axis of rotation but some distance apart, as shown in Fig. 3. Let the vertices of the triangles be linked by non-elastic constraints or chains which have some slack so that the triangles can rotate relative to each other before they become 'locked' at a constant relative displacement. When the frames are locked, the composition behaves like a bigger frame. This can be modeled as either a composite frame with four rails or as two single frames plus a disturbance term representing the tension of the connection chains. For the purpose of controlling the frames we used the second model in the design of a completely decentralized robust controller.

In Section 2.2., the biggest stabilization region for one isolated frame was shown to be $[\theta_m, \theta_M]$. A similar static analysis can be applied to the composite two frame system in the case when the frames are locked. This gives $M < \frac{3}{2}m\ tan^2\Delta$ as a sufficient condition for the new limit angle

$$\theta_{M_2} = arctg\Big(\frac{tan\frac{\alpha}{2} + tan\Delta\left(1+\frac{2M}{3m}\right)}{1+\frac{2M}{3m} - tan\frac{\alpha}{2}tan\Delta + \frac{2M}{3mcos\Delta}}\Big)$$

to be bigger than the one in (4). Thus, if this condition is satisfied, the composite system of two loosely coupled inverted pendulums have the potential to cooperate and operate over a zone of stability wider than that for the two pendulums operating in isolation.

5. Forming a production-line style composite system

Let the two loosely coupled triangular frame inverted pendulums plus associated product manipulators be configured into a production-line style system as shown in Fig. 4. The task of the system is to move a product from a "feeder" conveyor to an "exit" conveyor via an arbor on each of the triangular frames.

To transfer a product from the "feeder" conveyor into the arbor on frame $T1$, the arbor has to be aligned with the conveyor on which the product is waiting to be loaded (*loading point* θ_l). Similarly, to transfer a product from an arbor on frame $T2$ to the "exit" conveyor, frame $T2$ must be aligned with the conveyor at the (*unloading point* θ_u). For demonstration purposes, the arbors on Frames $T1$ and $T2$ are purpose designed to be asymmetric and are arrange such that stability zone for each frame includes only one of the load position and unload position (i.e. $T1$ will topple if moved to the unload position, and $T2$ will topple if moved to the load position). Thus, the production-line task can only be accomplished if both inverted pendulums cooperate such that $T1$ loads the product from the feeder conveyor and unloads the product to $T2$, and $T2$ loads the product from $T1$ and unloads the product to the exit conveyor. The point where the two frames exchange the part is called the *rendez-vous* point and is not a fixed position (although clearly the arbors must remain aligned throughout the transfer).

5.1. Task sequences for frames $T1$ and $T2$

For the triangular inverted pendulums $T1$, $T2$, the task sequence consists of issuing a sequence of control objectives:

Load_$T1$ if $T1$ not loaded, $T1$ go to loading point → the control objective is to move $T1$ to the set point θ_l using an appropriate motion profile and then for $T1$ to be held stationary at θ_l during the loading period;

Unload_$T1$ if $T1$ loaded, $T1$ go to rendez-vous point → the control objective is to move $T1$ (using appropriate positional and velocity control) such that it rendez-vous with $T2$ by aligning its arbor with that of $T2$ and then follows it with zero relative velocity for the period of the product transfer from $T1$ to $T2$ (i.e. unload $T1$).

Load_$T2$ if $T2$ not loaded, $T2$ go to rendez-vous point → the control objective is to move $T2$ (using appropriate positional and velocity control)such that it rendez-vous with $T1$ by aligning its arbor with that of $T1$ and then follows it with zero relative velocity for the period of the product transfer from $T1$ to $T2$ (i.e. load $T2$).

Unload_$T2$ if $T2$ loaded, $T2$ go to unloading point → the control objective to move $T2$ to the set point θ_u using an appropriate motion profile and then for $T2$ to be held stationary at θ_u during the unloading period;

The tasks sequence control objectives are translated in a series of different set points for the frames. Indicating with θ_1 the angular position of $T1$ and with θ_2 the $T2$ one

$$S(\text{Load\_}T1) = \dot{\theta}_1 + c(\theta_1 - \theta_l) \qquad (10)$$

$$S(\text{Unload\_}T1) = \dot{\theta}_1 + c(\theta_1 - \theta_2) \qquad (11)$$

$$S(\text{Unload\_}T2) = \dot{\theta}_2 + c(\theta_2 - \theta_u) \qquad (12)$$

$$S(\text{Load\_}T2) = \dot{\theta}_2 + c(\theta_2 - \theta_1) \qquad (13)$$

5.2. Coordination and synchronisation logic

The above rules define, in general terms, the logic necessary to coordinate and synchronise the frames and forms the functional requirement of the discrete event part of the hybrid system. Traditionally such logic would have been embedded as switching functions in the continuous controller. However, if the sensor and actuator interfaces are modelled at an abstract level they can be incorporated directly in the discrete event part of the system. To facilitate analysis and reasoning, the interfaces and discrete event system were modelled using control Petri nets which have a tangible graphical representation and an underlying mathematical structure.

Using control Petri nets the task sequences were modelled and the coordination and synchronisation logic designed as shown in Fig. 5. Transducers are modelled using a □ symbol and an arc leading to a transition: when a signal from the continuous layer is received the event "tokenises" the square and thus enables the transition (assuming all input places are tokenised). Actuators or continuous layer controllers are modelled by the symbol ◊ and an arc leading from a transition: when the transition fires ("tokenises" the diamond) it causes a command to be sent to the continuous layer actuator or controller. In the continuous layer a matching interface layer receives named signals from the discrete layer and translates them into real valued inputs for the continuous layer. Detailed control implementation features such as switches between the different set points, motion profiles, control modes or controller parameters, were embedded as lower level features within the continuous layer.

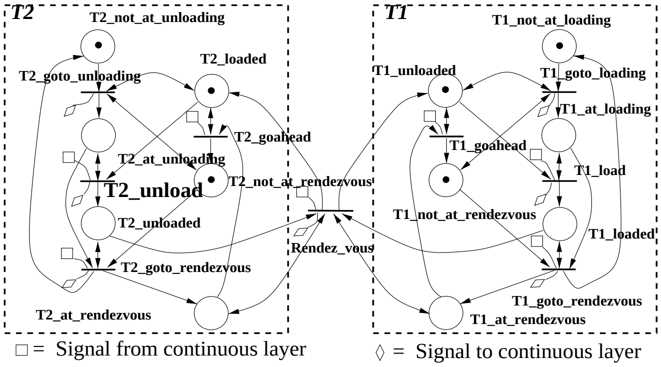


Figure 5:

5.3. Analysis and verification of synchronisation logic

The Petri net of the task sequence, shown in Fig. 5, consists of two components $T1$ and $T2$, which are synchronised via the transition *Rendez_vous*, which denotes the action of exchanging the component from one triangle to the other. The production–line sequence starts with $T1$ not loaded and located neither at the loading point nor at the Rendez_vous point. Firing of transition *T1_goto_loading* point sends $T1$ to the loading point (*T1_at_loading* is marked) and sends a signal to the continuous layer to implement the action. On receipt of the sensor signal indicating that $T1$ is at the loading point, *T1_load* fires and issues a command to the continuous layer to say that the component has to be loaded. Next firing of *T1_goto_rendezvous* sends the frame $T1$ to the rendez-vous point and since the triangle is not yet at the loading point, *T1_not_at_loading* is marked. By synchronisation via *Rendez_vous* transition the component is unloaded from $T1$ and transferred to $T2$ and a new cycle of operation starts for $T1$. For side $T2$ of the Petri net, the behaviour is dual of that of $T1$, as shown in Fig. 5. The Petri net of Fig. 5 is live (free from deadlocks) and safe (has a single, realisable, instance of states) and the reachability graph has 24 states and 41 arcs.

6. Cooperative behaviour in abnormal circumstances: the rescue function

In this section, we introduce the possibility that one of the triangular frames can go past the stability limit angle, perhaps due to an external impulsive disturbance, and start to 'tumble or fall down' (i.e. moves out of the stability zone and starts heading towards the stable equilibrium point for the uncontrolled system where it would dangling from the pivot). Because the loosely-coupled triangular inverted pendulums are linked by "chains", it follows that when one triangle begins to fall, the chains would tighten and the triangles would become locked and both would then fall. It follows that the chains provide a mechanism for the "domino" collapse of the system.

However simple appraisal shows that, if the two frames are identical and the ratio between m and M is suitable, the chains provide a mechanism that allows the non-falling triangular frame to come to the rescue of the falling frame. For example, given a method of detecting the onset of "falling" the non-falling triangular pendulum could throw itself in the opposite direction in the hope that the chains might prove to be "safety chains" and rescue the falling pendulum.
At this stage, the following *Falling policy* was introduced,

> If $T1$ ($T2$) is outside the region $[\theta_m, \theta_M]$, (i.e. $T1$ ($T2$) is falling), then $X1$ ($X2$) and $Y1$ ($Y2$) should be moved as fast as possible to $(x, y) = (0, 0)$ (the mechanical constraints imply that $(\dot{x}, \dot{y}) = (0, 0)$)

along with a *Rescue policy*,

> If $T1$ ($T2$) is outside the region $[\theta_m, \theta_M]$, (i.e. $T1$ ($T2$) is falling), then $T2$ ($T1$), should change its target set-point to a predetermined *rescue set-point*; when $T1$ ($T2$) is rescued, $T2$ ($T1$) should go back to try to reach the abandoned target set-point.

It is easy to see that the policy introduced for the single frame in Section 3.1., in which X and Y were sent to $(0, 0)$ when outside the stabilizability sector (to minimize the gravitational torque of the falling frame) turns out to be useful for the falling policy. Indeed, the regulator applied to the frames is exactly the same as before. Since one of the properties of the sliding mode control is to guarantee invariance of the behaviour of bounded disturbances of the control system, the current control law can ensure that the dynamic of the single frame is not affected by the disturbance imposed by the other frame.

6.1. Production-line style composite system with Rescue Mode

As a final phase of the design process, the falling/rescue policy was integrated with the task sequence. The resulting Petri net, Fig. 6, has been designed for the discrete part of the hybrid controller. With respect to the Petri Net in Fig. 5, the shaded places have been added along with new transitions. Place $p7$ is marked when none of the frames is in rescue mode. As soon as one among the transitions $t82$, $t92$, $t81$, $t91$ fires because a 'falling down' signal is received from the continuous layer, the token in $p7$ is removed, then no transition in the sub-nets linked to the regular behaviour can fire. Also the tokens in the places $p12$ (*T2_not_at_unloading*), $p11$ (*T1_not_at_loading*), $p22$ (*T2_at_unloading*), $p21$ (*T1_at_loading*), $p52$ (*T2_not_at_rendezvous*), $p51$ (*T1_not_at_rendezvous*), $p62$ (*T2_at_rendezvous*), $p61$ (*T1_at_rendezvous*), are temporarily removed because the conditions are no longer making sense during a rescue. Later, when the falling frame is safe, the two subnet will be restarted putting back the tokens in $p7$, $p12$, $p52$, $p11$, $p51$, and tokens are also added the transitions $t52$ and $t51$ to force $T2$ to go to the rendez-vous point if it is unloaded (in the original Petri Net the marking $\{p12, p32, p52, \cdot\}$ gives rise to deadlock), and to force $T1$ to go to the rendez-vous point if it is loaded, respectively.

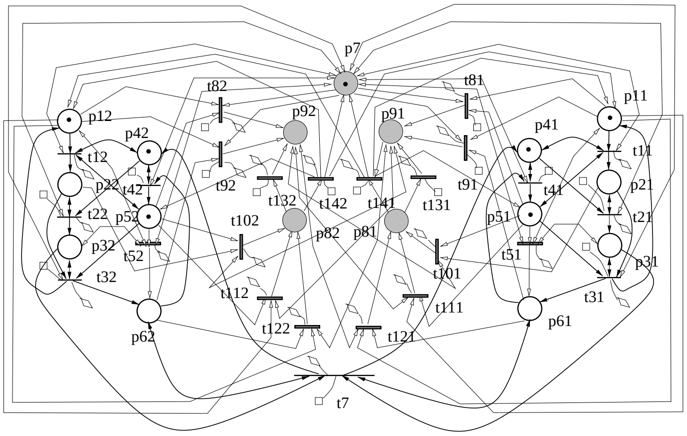


Figure 6:

When one of the two frames is falling down one of $p91$ and $p92$ is marked. Let's say that $T2$ is falling. Then, while tokens in $p12$, $p22$, $p52$ are removed from the transitions $t82$ or $t92$, and tokens in the $T1$ sub-net are removed from the only transitions enabled to fire (i.e. one among $t101$, $t111$, $t121$), and place $p81$ is marked, indicating $T1$ in rescue mode. At this stage one of two things can happen: $T2$ comes back to the stabilizability zone (it is said that it has been rescued) and both frames go back to the original tasks (transition $t141$) or $T1$ also goes in the unsafe sector and both frames will fall down (transition $t131$).

Analysis of the Petri net of Fig. 6, made with *Design/CPN_3.1.2* (University of Aarhus, Denmark), shows that its reachability graph has 35 states and 103 arcs. It has 4 deadlock configurations, each of which contains the marked places $p91$ and $p92$ which correspond, as expected, to the falling states of the two pendulums.

Analysis shows that the Petri net contains a 'livelock' or 'dynamic deadlock'. This corresponds to the firing sequence $\{t82, t111, t141, t81, t112, t142\}$, in which the two frames are cycling between the falling and rescuing states.

Verification of the system's function also involved searching for any transition firing sequences not containing the transitions $t7$ (*rendez-vous*), which indicates an operational deadlock due to geometric and physical constraints. The Petri net was found to contain such sequences and therefore pre–emption of such a sequence was used to ensure correct operation.

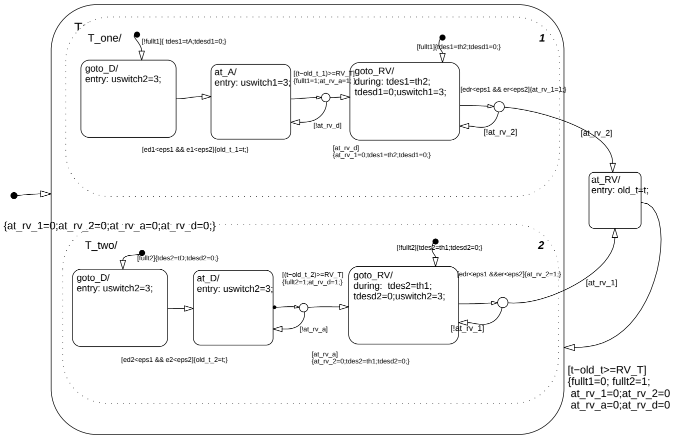


Figure 7:

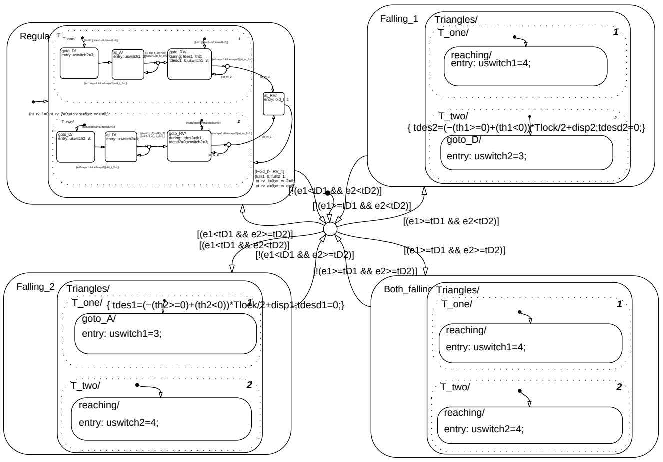


Figure 8:

7. Implementation with Matlab Toolbox

The continuous system has been implemented as a Simulink (Vers.3) model and comprises models of the two triangular inverted pendulums, the balance weight mechanisms, and the control system. The models of the inverted pendulum have been simplified in the locked case only to a composite pendulum model, but particular care has been devoted to the modeling of the lock/unlock conditions that permits switching between the composite model and the two separated models. The control system model consists of a set of simple regulators and lower level mode control switches.

The controlling Petri net has been translated into a Stateflow (Vers.2) diagram. This sends commands (i.e. signals) to the Simulink model (to select between motion profiles and control algorithms) in the form of numeric outputs that can assume values in a predefined set. It also receives events (i.e. signals) from the Simulink model. Since Stateflow is not a verifiable tool, the translation from the Petri net to Statechart [4] was achieved by designing a Statechart which implements the Petri net reachability graph.

In Fig. 8, there is the complete statechart with the four possible states: *Regular Motion*, *Falling_1* ($T1$ falling down), *Falling_2* ($T2$ falling down), *Both falling*. A test condition is used to switch between the four states at each time step. Going down in the hierarchy, in Fig. 7, there is the enlargment of the part of the chart connected to the regular motion phase. The dotted smooth rectangles indicates two parallel sections. They correspond to the $T1$ and $T2$ control that run concurrently. Outside the parallel boxes, the state *at_RV* can be seen: it is the synchronisation stage at the *rendez vous* point. Extensive testing at limits of working conditions has shown the correspondence between the expected behaviour and the implemented one. Figures 9 and 10 show the angular position and the angular velocities, respectively, of the arbors of the two synchronized frames for a regular sequence of movements. In Fig. 11 are shown the arbor trajectories for the case that triangle $T1$ starts in an unstable position and the lock angle is too big. The policy is that the rescueing frame stops the rescue phase as soon as the other frame is in a stable zone. The rescue takes place by $T2$ going to a rescue set-point placed within the stabilizability zone. In this case, when $T2$ reaches the rescue set-point, as the chains are too long, $T1$ is not in the safe area and the system deadlocks in a rescueing phase. In figure 12, $T2$ starts in an unstable position but the locking angle is less than $\frac{\alpha}{2}$. In figure 13, $T2$ the lock angle is too small to allow the schedulated task to be completed (in fact the angle between loading point and unloading point is bigger than the lock angle), then the two frames go to an equilibrium position in which the two arbors are at the minimal distance achievable.

8. Conclusions

In this paper, the application of Petri nets as a design tool for complex supervisory layers in a hybrid environment has been shown through an example. The overall design is clear and easily understood; it is analysable in a structured way, and maintenable. Making the decision layer amenable to analysis provides invaluable feedback concerning potential behaviour. This provides information that can be used to reshape control parameters or the structural parameters of the continuous system. Also, timing information derived from the continuous system can be added to the Petri net to form a time Petri net that can be analyzed to determine whether the time constraints prohibit unwanted behaviour, or introduce other problems such as other deadlocks.

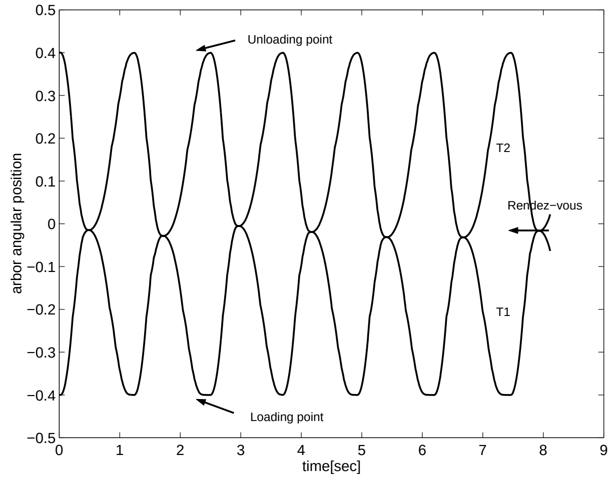


Figure 9: T_1 arbor angle (with respect to the center of the frame)$= 0.4$, T_2 arbor angle $= -0.4$, $\theta_1(0) = \dot{\theta}_1(0) = \theta_2(0) = \dot{\theta}_2(0) = 0$, $\alpha_{lock} = \frac{\pi}{2}$, $\alpha = \pi$.

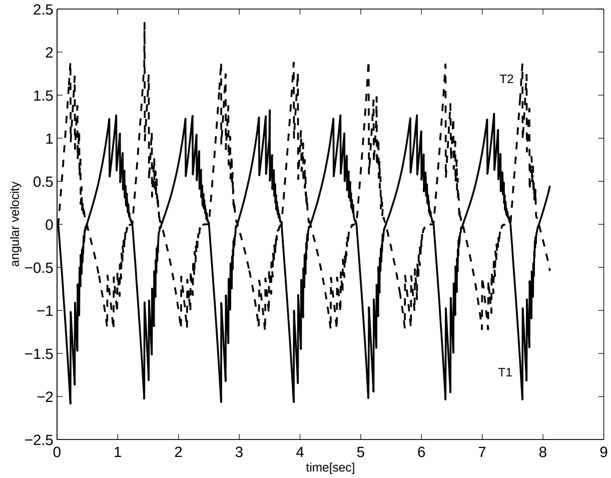


Figure 10: T_1 arbor angle (with respect to the center of the frame)$= 0.4$, T_2 arbor angle $= -0.4$, $\theta_1(0) = \dot{\theta}_1(0) = \theta_2(0) = \dot{\theta}_2(0) = 0$, $\alpha_{lock} = \frac{\pi}{2}$, $\alpha = \pi$.

Equally valuable, the reachability tree allows backtracking from deadlock or undesirable states and allows precursor behaviour to be detected and preempting actions to be designed.

Acknowledgements The current work has been supported by (UK) EPSRC Grant GR/L31234.

References

[1] Bartolini, G., Ferrara, A., Usai, E.: Applications of a suboptimal discontinuous control algorithm for uncertain second order systems. Int. J. of Robust Nonlin. Control, vol. 7, pp. 299-320 (1997)

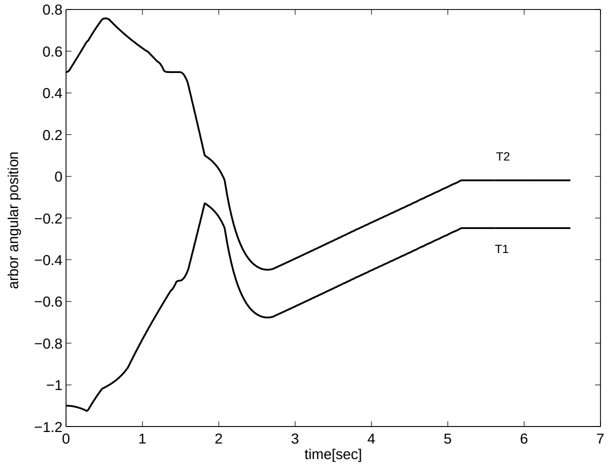


Figure 11: T_1 arbor angle (with respect to the center of the frame)$= 0.5$, T_2 arbor angle $= -0.5$, $\theta_1(0) = 0.6$, $\dot{\theta}_1(0) = -1.$, $\theta_2(0) = \dot{\theta}_2(0) = 0$, $\alpha_{lock} = 1.2$, $\alpha = \pi$.

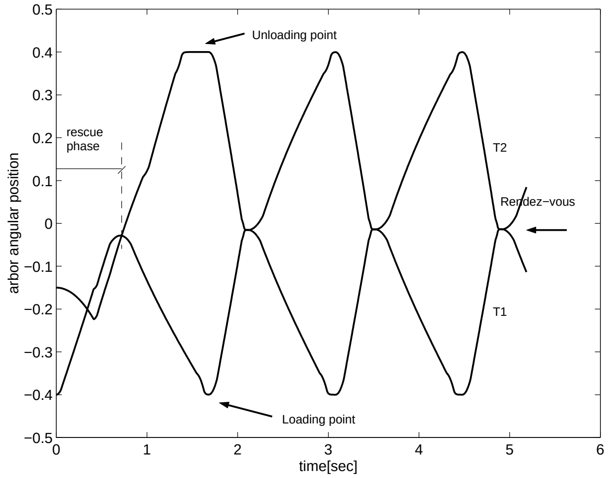


Figure 12: T_1 arbor angle (with respect to the center of the frame)$= 0.4$, T_2 arbor angle $= -0.4$, $\theta_1(0) = \dot{\theta}_1(0) = \dot{\theta}_2(0) = 0$, $\theta_2(0) = -0.55$, $\alpha_{lock} = \frac{\pi}{2}$, $\alpha = \pi$.

[2] Cassandras, C.G.: Discrete Event Systems : Modeling and Performance Analysis. Irwin Publ. (1993)

[3] Grossman, R.L., Nerode, A., Ravn, A.P., Rischel, H.(eds.): Hybrid Systems. Lecture Notes in Computer Science, Vol. 736. Springer–Verlag (1993)

[4] Harel, D.: Statecharts: A Visual Formalism for Complex Systems. Science of Computer Programming, Vol. 8 (1987) 231–274

[5] Koutsoukos, X.D., Antsaklis, P.J.:Hybrid Control Systems Using Timed Petri Nets: Supervisory ControlDesign Based on Invariant Properties. Hybrid Systems V, P. Antsaklis, W. Kohn, M. Lemmon, A. Nerode,

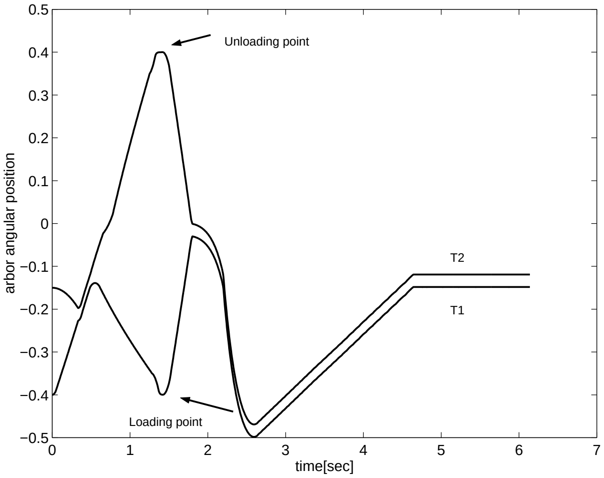


Figure 13: T_1 arbor angle (with respect to the center of the frame)= 0.4, T_2 arbor angle = -0.4, $\theta_1(0) = \dot{\theta}_2(0) = 0$, $\dot{\theta}_1(0) = -1.$, $\theta_2(0) = -0.55$, $\alpha_{lock} = 0.6$, $\alpha = \pi$.

S. Sastry Eds.. Lecture Notes in Computer Science, LNCS 1567, Springer-Verlag (1999)

[6] Murata, T.: Petri-Nets: Properties, Analysis and Applications. Proceedings of the IEEE, Vol. 77 (1989) 541–580

[7] Utkin, V.I.: Sliding Modes In Control And Optimization. Berlin: Springer Verlag (1992)