A Simple Hybrid Event-B Model of an Active Control System for Earthquake Protection

Richard Banach and John Baugh

Abstract In earthquake-prone zones of the world, severe damage to buildings and life endangering harm to people pose a major risk when severe earthquakes happen. In recent decades, active and passive measures to prevent building damage have been designed and deployed. A simple model of an active damage prevention system, founded on earlier work, is investigated from a model based formal development perspective, using Hybrid Event-B. The non-trivial physical behaviour in the model is readily captured within the formalism. However, when the usual approximation and discretization techniques from engineering and applied mathematics are used, the rather brittle refinement techniques used in model based formal development start to break down. Despite this, the model developed stands up well when compared via simulation with a standard approach. The requirements of a richer formal development framework, better able to cope with applications exhibiting non-trivial physical elements are discussed.

1 Introduction

In earthquake-prone zones of the world, damage to buildings during an earthquake is a serious problem, leading to major rebuilding costs if the damage is severe. This is to say nothing of the harm to people that ensues if they happen to be inside, or near to, a building that fails structurally. One approach to mitigating the problem is to make buildings so robust that they can withstand the severest earthquake that may befall them; but this not only greatly increases cost, but also places limits on

Richard Banach
School of Computer Science, University of Manchester, Oxford Road, Manchester, M13 9PL, U.K., e-mail: richard.banach@manchester.ac.uk

John Baugh
Department of Civil, Construction, and Environmental Engineering, North Carolina State University, Raleigh, NC 27695-7908, USA, e-mail: jwb@ncsu.edu

the size and shape of buildings, so that the desired robustness remains feasible with available materials.

In recent decades, an alternative approach to earthquake protection has been to use control techniques to dissipate the forces that reach vulnerable elements of a building by using control strategies of one kind or another. In truth, the first proposal for intervening in a building's ability to withstand earthquake dates back to 1870 (a patent filed at the U.S. Patent Office by one Jules Touaillon), but such ideas were not taken seriously till a century or so later.

One approach is to use passive control. In this approach massive members and/or damping mechanisms are incorporated into the building in such a way that their parameters and the coupling between them and the rest of the building are chosen just so that that the destructive forces are preferentially dissipated into these additional elements, leaving the building itself undamaged.

An alternative, more recent approach, is to use active control. Since it is the amplitude and frequency of the vibrations that a building is subject to during an earthquake that determine whether it will sustain damage ot not, damping earthquake vibrations by applying suitably designed counter vibrations to the building reduces the net forces that the building must withstand, and thus the damage it will sustain under a given severity of earthquake and given a specific standard of construction.

In [31, 16, 20, 21] there is a study of such an active control system for earthquake resistance for buildings. Ultimately, it is targetted at an experimental tall building of six stories. These papers investigate various aspects of verification for a system of this kind, based largely on timing considerations, which inevitably generate uncertainties due to equipment latencies. The approach to the system is rather bottom up in [20, 21]. The design is presented at a low level of detail with separate elements for the start of an action, the end of the action, and a synchronisation point within the action (as needed), with timings attaches to each element. Even for a simple system, this results in a huge state space. The focus of [20, 21] then becomes reduction of the state space size, showing no loss of behaviour via bisimulation. Following this, useful properties of the system model may be demonstrated using the smaller state space version.

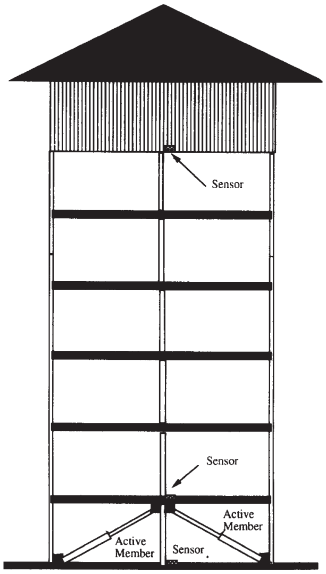


Fig. 1 A schematic of a building design, to be protected by an active earthquake damage prevention system. From [20, 21].

In the present paper, we take an alternative route, going top down instead of bottom up, and using Hybrid Event-B (henceforth HEB) [11, 12] as the vehicle for doing the development. We work top-down, and for simplicity and through lack of space, we do not get to the low level of detail present in [20, 21]. In particular, we omit replication of subsystems, timing and fault tolerance (though we comment on these aspects at the end). As well as providing the contrast to the previous treatment, the present case study offers some novelty regarding the interaction of physical and digital behaviours (in particular, regarding continuous vs. impulsive physics, as treated within the HEB formalism), compared with other case studies, e.g. [7, 9, 10, 5, 6, 15].

The rest of this paper is as follows. In Section 2 we briefly overview control strategies for seismic protection for buildings, and focus on the active control principles that underpin this paper's approach. Section 3 has an outline of single machine HEB, for purposes of orientation. In Section 4 we present the simple dynamical model we develop, and its most abstract expression in HEB. and Section 5 presents an ideal but completely unrealistic solution to the problem posed in the model. The next few sections develop and refine the original model in a less idealised way, bringing in more of the detailed requirements of a practical solution. The more detail we bring in, the greater the challenge to the usual refinement technique found in formal development frameworks, including HEB.

Thus Section 6 presents a first refinement towards a practical solution, while Section 7 engages more seriously with an 'ideal pulse' strategy for the active control solution. Section 8 pauses to discuss the issues for refinement that this throws up. Section 9 incorporates the discretization typically seen in practical engineering solutions, and also treats decomposition into a family of machines that reflect a more convincing system architecture, one resembling the approach of [20, 21]. Section 10 continues the discussion of issues raised for refinement and retrenchment by these development steps. Section 11 presents numerical simulation work showing that the theoretically based earlier models give good agreement when compared with solutions derived using conventional engineering approaches. Section 12 recapitulates and concludes.

2 Control Strategies for Earthquake Damage Prevention

Since mechanical prevention of earthquake damage to structures began to be taken seriously, a number of engineering techniques have been brought to bear on the problem [40]. These days this is a highly active field and the literature is large, e.g. [1, 2, 19].

In passive control [17], decoupling of the building from the ground, and/or the incorporation of various additional members, are used to ensure that the forces of an earthquake do not impinge on the important building structure. Passive approaches are often used to protect historical buildings in which major re-engineering is impractical. One disadvantage of the passive approach is the potential transverse displacements relative to the ground that the protected building may undergo. If, with respect to an inertial frame, the building stays still, and the ground moves by 20cm., then the relative movement of the building is 20cm. This may not be practical.

In alternative approaches the engineering compensation is more active. Active approaches (such as the one we will pursue in more detail below) have the compensation mechanism trying to actively counter the forces imparted by the earthquake in order to limit the amplitude of vibrations at the building's resonant frequencies [33, 34]. One problem experienced by active prevention systems is that they may consume a lot of energy, which is expensive and undesirable. Another is that if one is unlucky, and the parameters of the earthquake fall in the wrong region (due to imprescient design, or error), then because an active system is injecting energy into the overall structure, it may actually make things worse, driving the overall structure into instability, perhaps because the injected energy is being introduced in phase rather than in anti-phase with the earthquake itself. An increasingly popular approach these days is semi-active control [22], in which the main aim is to intervene actively only in the dissipative part of the building's response, decreasing energy costs and avoiding potential instabilities, for only a small decrease in performance.

In all of these active strategies for prevention of earthquake damage to buildings, the building contains a set of sensors which constantly monitor vibrations impinging on the building from the ground. The signals coming from these are analysed to differentiate between earthquake originated forces, and normal day to day vibrations caused by everyday activities in the building's surroundings. The latter are, of course, ignored.

The building also contains active members which can impart forces to the building structure. The aim of the active control is to impart forces to the building that counter the damaging forces coming from the earthquake, so that the net force that the building must withstand remains within its safe design parameters.

These days, these design aims are achieved using a sophisticated control engineering approach. Many strategies have been tried, but among the most popular currently is to use a LQG (Linear Quadratic Gaussian) strategy for designing a nominal controller, which is then modulated by clipping extreme values. This approach is based on a sophisticated formulation of noisy dynamics and its control via a Bayesian estimation of current and future behaviour. See e.g. [22] (or [28] for a simpler example). One consequence of this approach is some loss of direct contact between the control algorithm design and real time values, due to the use of L^2 estimates in the derivation of the controller. This is a disadvantage regarding direct correspondence with typical formal methods approaches, which are wedded exclusively to variable values in real time.

Our own study is based on the strategy used in [20, 21]. Fig. 1, taken from [20, 21], gives a schematic outline of how active elements are disposed in an experimental building in Tokyo. There are sensors near the ground, and at the top of the building. The active members, which have to be capable of exerting significant force if they are to move significant aspects of the building, are found at the bottom of the building.[1]

The technique by which the corrective forces are applied to the building is to have the active members impart a series of pulses to the core framework of the building. Of course, for this to be successful on the timescales of earthquake vibra-

[1] In sophisticated modern designs, active members are also found higher up the building, to counter vibration antinodes part way up a tall structure.

```
MACHINE $HyEvBMch$
TIME $t$
CLOCK $clk$
PLIANT $x, y$
VARIABLES $u$
INVARIANTS
  $x \in \mathbb{R}$
  $y \in \mathbb{R}$
  $u \in \mathbb{N}$
EVENTS
  INITIALISATION
    STATUS ordinary
    WHEN
      $t = 0$
    THEN
      $clk := 1$
      $x := x_0$
      $y := y_0$
      $u := u_0$
    END
... ...
```

```
... ...
  $MoEv$
    STATUS ordinary
    ANY $i?, l, o!$
    WHERE $grd(x, y, u, i?, l, t, clk)$
    THEN
      $x, y, u, clk, o! :| BApred(x, y, u,$
        $i?, l, o!, t, clk, x', y', u', clk')$
    END
  $PliEv$
    STATUS pliant
    INIT $iv(x, y, t, clk)$
    WHERE $grd(u)$
    ANY $i?, l, o!$
    COMPLY $BDApred($
      $x, y, u, i?, l, o!, t, clk)$
    SOLVE
      $\mathcal{D}\, x = \phi(x, y, u, i?, l, o!, t, clk)$
      $y := E(x, y, u, i?, l, o!, t, clk)$
    END
END
```

Fig. 2 A schematic Hybrid Event-B machine.

tions, there has to be accurate real time control, and an appropriate balance between the aggregated effect of the applied pulse series and the sensed vibrations coming from the earthquake. In contrast to the LQG approach, the technique used in [29, 31, 16, 20, 21] is based on real time monitoring of positions, velocities and accelerations in the building's structure, thus greatly facilitating a correspondence with conventional model based formal methods techniques (a point that emerges, though obviously quite indirectly, from remarks in [29]).

It is not our aim in this paper to get deeply embroiled in the detailed control engineering aspects of the problem. We leave that to other work. Instead, our aim is to take a top down approach to the implementation task, and to see how a HEB perspective can bring efficiencies and a degree of clarity to that. Accordingly, we next turn to HEB itself.

3 A Brief Outline of Hybrid Event-B

In this section we give an outline of Hybrid Event-B for single machines. In Fig. 2 we see a bare bones HEB machine, *HyEvBMch*. It starts with declarations of time and of a clock. In HEB, time is a first class citizen in that all variables are functions of time, whether explicitly or implicitly. However time is special, being read-only, never being assigned, since time cannot be controlled by any human-designed engineering process. Clocks allow a bit more flexibility, since they are assumed to increase their value at the same rate that time does, but may be set during mode events (see below). Variables are of two kinds. There are mode variables (like u, declared as usual) which take their values in discrete sets and change their values via discontinuous assignment in mode events. There are also pliant variables (such as x,y), declared in the PLIANT clause, which take their values in topologically dense sets (normally $\mathbb{R}$) and which are allowed to change continuously, such change being specified via pliant events (see below).

Next are the invariants. These resemble invariants in discrete Event-B, in that the types of the variables are asserted to be the sets from which the variables' values *at any given moment of time* are drawn. More complex invariants are similarly predicates that are required to hold *at all moments of time* during a run.

Then we get to the events. The *INITIALISATION* has a guard that synchronises time with the start of any run, while all other variables are assigned their initial values in the usual way. As hinted above, in HEB, there are two kinds of event: mode events and pliant events.

Mode events are direct analogues of events in discrete Event-B. They can assign all machine variables (except time itself). In the schematic *MoEv* of Fig. 2, we see three parameters $i?,l,o!$, (an input, a local parameter, and an output respectively), and a guard *grd* which can depend on all the machine variables. We also see the generic after-value assignment specified by the before-after predicate $BApred$, which can specify how the after-values of all variables (except time, inputs and locals) are to be determined.

Pliant events are new. They specify the continuous evolution of the pliant variables over an interval of time. The schematic pliant event *PliEv* of Fig. 2 shows the structure. There are two guards: there is *iv*, for specifying enabling conditions on the pliant variables, clocks, and time; and there is *grd*, for specifying enabling conditions on the mode variables. The separation between the two is motivated by considerations connected with refinement.

The body of a pliant event contains three parameters $i?,l,o!$, (once more an input, a local parameter, and an output respectively) which are functions of time, defined over the duration of the pliant event. The behaviour of the event is defined by the COMPLY and SOLVE clauses. The SOLVE clause specifies behaviour fairly directly. For example the behaviour of pliant variable y is given by a direct assignment to the (time dependent) value of the expression $E(\ldots)$. Alternatively, the behaviour of pliant variable x is given by the solution of the first order ordinary differential equation (ODE) $\mathcal{D}x = \phi(\ldots)$, where $\mathcal{D}$ indicates differentiation with respect to time. (In fact the sematics of the $y = E$ case is given in terms of the ODE $\mathcal{D}y = \mathcal{D}E$, so that both x and y satisfy the same regularity properties.) The COMPLY clause can be used to express any additional constraints that are required to hold during the pliant event via its before-during-and-after predicate $BDApred$. Typically, constraints on the permitted range of values for the pliant variables, and similar restrictions, can be placed here.

The COMPLY clause has another purpose. When specifying at an abstract level, we do not necessarily want to be concerned with all the details of the dynamics — it is often sufficient to require some global constraints to hold which express the needed safety properties of the system. The COMPLY clauses of the machine's pliant events can house such constraints directly, leaving it to lower level refinements to add the necessary details of the dynamics.

Briefly, the semantics of a HEB machine is as follows. It consists of a set of *system traces*, each of which is a collection of functions of time, expressing the value of each machine variable over the duration of a system run. (In the case of *HyEvBMch*, in a given system trace, there would be functions for clk, x, y, u, each defined over the duration of the run.)

Time is modeled as an interval $\mathcal{T}$ of the reals. A run starts at some initial moment of time, t_0 say, and lasts either for a finite time, or indefinitely. The duration of the run $\mathcal{T}$, breaks up into a succession of left-closed right-open subintervals: $\mathcal{T} = [t_0 \ldots t_1), [t_1 \ldots t_2), [t_2 \ldots t_3), \ldots$. The idea is that mode events (with their discontinuous updates) take place at the isolated times corresponding to the common endpoints of these subintervals t_i, and in between, the mode variables are constant and the pliant events stipulate continuous change in the pliant variables.

Although pliant variables change continuously (except perhaps at the t_i), continuity alone still allows for a wide range of mathematically pathological behaviours. To eliminate these, we make the following restrictions which apply individually to every subinterval $[t_i \ldots t_{i+1})$:

I Zeno: there is a constant δ_{Zeno}, such that for all i needed, $t_{i+1} - t_i \geq \delta_{\mathsf{Zeno}}$.

II Limits: for every variable x, and for every time $t \in \mathcal{T}$, the left limit $\lim_{\delta \to 0} x(t - \delta)$ written $\overrightarrow{x(t)}$ and right limit $\lim_{\delta \to 0} x(t + \delta)$, written $\overleftarrow{x(t)}$ (with $\delta > 0$) exist, and for every t, $x(t) = \overleftarrow{x(t)}$. [N. B. At the endpoint(s) of $\mathcal{T}$, any missing limit is defined to equal its counterpart.]

III Differentiability: The behaviour of every pliant variable x in the interval $[t_i \ldots t_{i+1})$ is given by the solution of a well posed initial value problem $\mathcal{D}\, xs = \phi(xs \ldots)$ (where xs is a relevant tuple of pliant variables and $\mathcal{D}$ is the time derivative). 'Well posed' means that $\phi(xs \ldots)$ has Lipschitz constants which are uniformly bounded over $[t_i \ldots t_{i+1})$ bounding its variation with respect to xs, and that $\phi(xs \ldots)$ is measurable in t.

Regarding the above, the Zeno condition is certainly a sensible restriction to demand of any acceptable system, but in general, its truth or falsehood can depend on the system's full reachability relation, and is thus very frequently undecidable.

The stipulation on limits, with the left limit value at a time t_i being not necessarily the same as the right limit at t_i, makes for an easy interpretation of mode events that happen at t_i. For such mode events, the before-values are interpreted as the left limit values, and the after-values are interpreted as the right limit values.

The differentiability condition guarantees that from a specific starting point, t_i say, there is a maximal right open interval, specified by t_{MAX} say, such that a solution to the ODE system exists in $[t_i \ldots t_{\mathrm{MAX}})$. Within this interval, we seek the earliest time t_{i+1} at which a mode event becomes enabled, and this time becomes the preemption point beyond which the solution to the ODE system is abandoned, and the next solution is sought after the completion of the mode event.

In this manner, assuming that the *INITIALISATION* event has achieved a suitable intial assignment to variables, a system run is *well formed*, and thus belongs to the semantics of the machine, provided that at runtime:

- Every enabled mode event is feasible, i.e. has an after-state, and on its completion enables a pliant event (but does not enable any mode event).[2] (1)
- Every enabled pliant event is feasible, i.e. has a time-indexed family of after-states, and EITHER: (2)

  (i) During the run of the pliant event a mode event becomes enabled. It preempts the pliant event, defining its end. ORELSE
  (ii) During the run of the pliant event it becomes infeasible: finite termination. ORELSE
  (iii) The pliant event continues indefinitely: nontermination.

Thus in a well formed run mode events alternate with pliant events. The last event (if there is one) is a pliant event (whose duration may be finite or infinite).

We note that this framework is quite close to the modern formulation of hybrid systems. (See e.g. [35, 27] for representative formulations, or the large literature in the *Hybrid Systems: Computation and Control* series of international conferences, and the further literature cited therein.)

In reality, there are a number of semantic issues that we have glossed over in the framework just sketched. We refer to [11] for a more detailed presentation. Also, the development we undertake requires the multi-machine version of HEB [12]. Since the issues that arise there are largely syntactic, we explain what is needed for multi-machine HEB *in situ*, as we go along.

4 Top Level Abstract Model of the Control System

As stated in Section 2, we do not get deeply embroiled in the detailed control engineering aspects of realistic active control in this paper. We base our treatment on the relatively simple strategy described in detail in [31].

Fig. 3 shows the simple system investigated in [31, 29]. The model refers to the dynamics of a mechanical system with a single degree of freedom (SDOF). The building to be protected is modelled as a concentrated or lumped mass m and a structural system that resists lateral motion with a spring of stiffness k and a viscous damper with coefficient c.[3] A force p is applied to the mass by the active control system. The effects of spring and damper depend on the relative position between a fiducial point in the building w, and another fiducial point in the earth z, i.e. on

[2] If a mode event has an input, the semantics assumes that its value arrives at a time distinct from the previous mode event, ensuring part of (1) automatically.

[3] Idealizing a building by an equivalent SDOF system requires an assumption about its displaced shape and other details that are beyond the scope of the paper. The interested reader is directed to the methodology outlined by Kuramoto et al. [24], which is included in the current building design code used in Japan, as one example.

$x = w - z$. When $w = z = 0$, the spring is unstretched. Writing $\mathcal{D}$ for the time derivative, and defining $e \equiv \mathcal{D}^2 z$, the dynamics of w, expressed in terms of the relative displacement x, is thus controlled by:

$$m\mathcal{D}^2 x + c\mathcal{D}x + kx = p - me \tag{3}$$

Since p is to be chosen by the system, e can be measured, and the other data are known, (3) yields a yields a law that can be used to keep x within desired bounds.

The code for the top level model of the HEB development is in Fig. 4. At this level the system consists of a single machine, *ActConMch_0*. There are pliant variables x, p, which capture the model elements discussed above.

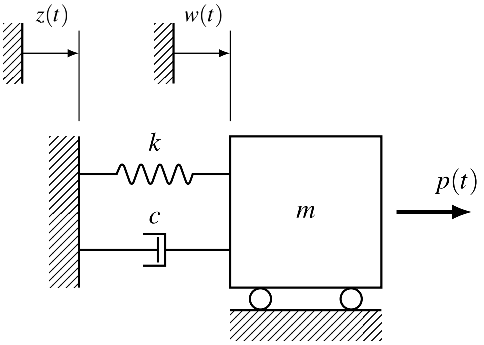


Fig. 3 The simple mechanical model that the HEB development is based on, after [31].

The *INVARIANTS* are rather basic at this stage. They declare the types of the variables, and one further non-trivial property. This property actually expresses the key system requirement of the whole development, namely that the value of variable x stays within a range $-X_B \leq x \leq X_B$. Imposing it amounts to placing a limit on the lateral drift[4] of a building structure, i.e., the horizontal displacement that upper stories undergo with respect to the base. This relative motion is resisted by the building's structural system, but under extreme events the internal deformations may be excessive, leading to structural damage and ultimately collapse of the building.

```
  MACHINE ActConMch_0
  PLIANT $x, p$
  INVARIANTS
    $x, p \in \mathbb{R}, \mathbb{R}$
    $|x| \leq X_B$
  EVENTS
    INITIALISATION
      STATUS ordinary
      BEGIN
        $x, p$ := $0, 0$
      END
... ...
```

```
... ...
    MONITOR
      STATUS pliant
      ANY $e?$
      WHERE
        $e? \in \mathbb{R} \wedge |e?| \leq E_B$
      COMPLY INVARIANTS
      END
  END
```

Fig. 4 A highly abstract model of the earthquake damage prevention active control system.

[4] The International Building Code (IBC) requires that drift be limited in typical buildings to 1–2% of the building's height for reasons of both safety and functional performance.

The *INITIALISATION* event sets all the variables to zero. Then there is the single actual event of the model: the *MONITOR* pliant event, which covers the continuous monitoring of the system when it is in the monitoring mode. Since this is the only mode in the model, it does not require a specific variable or value to name it.[5]

The definition of the *MONITOR* pliant event is trivial at this level of abstraction. It consumes its input $e?$, evidently corresponding to the relevant model element above. For future calculational tractability, $e?$ is assumed to be bounded by an explicit constant E_B. The event simply demands that the *INVARIANTS* are to be maintained. This is to be established in some as yet unspecified manner, since we postpone the more demanding details of the calculations involved in the control model we have introduced. Later on, its more precisely defined job will be to monitor the information coming from the sensors (i.e. to monitor w and its derivatives, and e), to calculate what response may be necessary, and to issue pulses to the actuators, as may be required, these being the embodiment of p. Of course, in reality, the monitoring will be done via a series of discrete events, resulting in a series of readings from the sensors, but for the next few models in the development, we will assume that this is all a continuous process.

5 An Idealised Refinement: Miraculous ODE Behaviour

Fig. 5 presents a somewhat idealised refinement of *ActConMch_0*. Rather than assume the desired effect is achieved nondeterministically, we introduce the control law (3) in the SOLVE clause of *MONITOR*. In order to do that we introduce a new variable y so that we can translate the second order (3) into the first order form stipulated by HEB. Having done that, we notice that we need merely to set p to $m\,e?$ and the zero initialisation of x,y persists to a global solution satisfying the invariants. We are done! The building stands still under all admissible earthquake conditions!

If only it were that simple. Unfortunately, it requires that p be chosen to mirror the instantaneous real time behaviour of $e?$ with complete precision, with no allowance for quantisation effects or for signal propagation delay in equipment. This is impractical in the real world. Accordingly, we abandon this route in favour of a more achievable development route.

6 A More Realistic Refinement: Achievable ODE Behaviour

The problem with the *MONITOR* of Fig. 5 is that it is already so precise that there is no way to backtrack to a more tolerant engineering model *while remaining within the restrictions of refinement theory*. In this section we have a different refinement of *ActConMch_0*, which allows some leeway for engineering imprecision.

[5] Of course, a more realistic model would contain modes for maintenance, and for other forms of partial running or of inactivity — to be used, presumably, only when the building is unoccupied.

```
MACHINE ActConMch_IDEAL
REFINES ActConMch_0
PLIANT $x, y, p$
INVARIANTS
  $x, y, p \in \mathbb{R}, \mathbb{R}, \mathbb{R}$
  $|x| \leq X_B$
EVENTS
  INITIALISATION
    REFINES INITIALISATION
    STATUS ordinary
    BEGIN
      $x, y, p := 0, 0, 0$
    END
... ...
```

```
... ...
  MONITOR
    REFINES MONITOR
    STATUS pliant
    ANY e
    WHERE
      $e? \in \mathbb{R} \wedge |e?| \leq E_B$
    SOLVE
      $p := me?$
      $\mathcal{D}x = y$
      $\mathcal{D}y = -\frac{c}{m}y - \frac{k}{m}x + \frac{1}{m}p - e?$
    END
END
```

Fig. 5 Idealised refinement of the system.

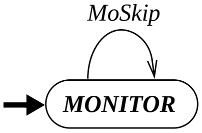


Fig. 6 A transition diagram for the first refinement of the HEB model of the earthquake damage prevention system.

In Fig. 6 we have a transition diagram representation of the system for this refinement. As before there is only one state, the one occupied by the *MONITOR* event, and there is now a mode event too, *MoSkip*, of which more later. The HEB code for the refinement is in Fig. 7.

The behaviour of the *MONITOR* event refines its previous incarnation by restricting the behaviour. As well as the input $e?$, there are now two locally chosen parameters, pp and e. The former allows values that match $e?$ imprecisely to be fed to the ODE system in the SOLVE clause (which is almost the same as in Fig 5), while the latter permits the stipulation that $e?$ differs from a constant value (which may be chosen conveniently) by not too much during a *MONITOR* transition.

The COMPLY clause takes advantage of the fact that the solution to such linear constant coefficient inhomogeneous ODE systems is routine. See [36, 32, 18, 4] as well as a host of other sources. The first term is the homogeneous solution, primed by the initial values: $\mathrm{e}^{\boldsymbol{A}(t-\mathbb{t}_{\mathrm{L}})}[x(\mathbb{t}_{\mathrm{L}}), y(\mathbb{t}_{\mathrm{L}})]^{\mathrm{T}}$, where $\boldsymbol{A}$ is the companion matrix of the homogeneous part of the ODE system in the SOLVE clause, and $\mathbb{t}_{\mathrm{L}}$ refers, generically, to the start time of any runtime transition specified by the pliant event. The second term is the convolution $*$ over the interval $[\mathbb{t}_{\mathrm{L}} \dots t]$ between the homogeneous solution $\mathrm{e}^{\boldsymbol{A}(s)}$ (with bound convolution variable renamed to s) and the inhomogeneous part $[0, \frac{1}{m}pp - e?]^{\mathrm{T}}$. If the projection of all this to the x variable (written $.x$) achieves the desired bound, then the ODE system in the SOLVE clause establishes the desired invariant. The permitted imprecision between pp and $e?$ now makes this a practical proposition.

The mode event *MoSkip* interrupts the *MONITOR* event at intervals of T_P, after which *MONITOR* restarts. This permits the reassignment of the constant e in *MONITOR* at each restart. If the interval T_P is short enough it permits the choice of pp during each *MONITOR* transition to achieve the desired outcome.

```
MACHINE  ActConMch_1
REFINES  ActConMch_0
CLOCK  clk_pls
PLIANT  $x, y, p$
INVARIANTS
  $x, y, p \in \mathbb{R}, \mathbb{R}, \mathbb{R}$
  $|x| \leq X_B$
EVENTS
  INITIALISATION
    REFINES  INITIALISATION
    STATUS  ordinary
    BEGIN
      $clk\_pls := 0$
      $x, y, p := 0, 0, 0$
    END
  MoSkip
    STATUS  anticipating
    WHEN
      $clk\_pls = T_P$
    THEN
      $clk\_pls := 0$
    END
...  ...
```

```
...  ...
  MONITOR
    REFINES  MONITOR
    STATUS  pliant
    ANY  $pp, e, e?$
    WHERE
      $pp \in \mathbb{R} \wedge |pp| \leq PP_B \wedge$
      $e \in \mathbb{R} \wedge \mathsf{CONST}(e) \wedge$
      $e? \in \mathbb{R} \wedge |e?| \leq E_B \wedge |e - e?| \leq e_B$
    COMPLY
      $\{\mathrm{e}^{\boldsymbol{A}(t - \mathbb{t}_\mathrm{L})}[x(\mathbb{t}_\mathrm{L}), y(\mathbb{t}_\mathrm{L})]^\mathrm{T} +$
          $(\mathrm{e}^{\boldsymbol{A}s} *_{[\mathbb{t}_\mathrm{L} \ldots t]} [0, \frac{1}{m}pp - e?]^\mathrm{T})\}.x$
      $\leq X_B$
    SOLVE
      $p := pp$
      $\mathcal{D}x = y$
      $\mathcal{D}y = -\frac{c}{m}y - \frac{k}{m}x + \frac{1}{m}p - e?$
    END
END
```

Fig. 7 First refinement of the system.

The caption of Fig. 7 claims that *ActConMch_1* is a refinement of *ActConMch_0*. Indeed it is, although we do not describe the details of this here; see [11] for a more thorough account. We note though that: the invariants are not weakened; the new initialisation is evidently consistent with the old; the new behaviour of *MONITOR* evidently satisfies the previous definition; and the new mode event only updates a newly introduced (clock) variable. All of these are characterisics of HEB refinement.

7 Refining pp

The objective of the next refinement is to address the specific form of pp, bringing the design closer to engineering practice, and to [31, 29] in particular. For this we follow the detailed formulation in [19], which contains a wealth of detailed calculation. We we make a conventional change of parameters: $\zeta = c/2\sqrt{km}$, $\omega_n = \sqrt{k/m}$, $\omega_D = \omega_n\sqrt{1 - \zeta^2}$. This change reduces the LHS of (3) to (m times):

$$\mathcal{D}^2 x + 2\zeta\omega_n \mathcal{D}x + \omega_n^2 x \tag{4}$$

In terms of these quantities, the generic solution of the ODE system indicated above reduces to a Duhamel integral with specified initial values [19, 39]:

$$
\begin{aligned}
x(t-\mathbb{t}_{\mathrm{L}}) \;=\;& \\
& \mathrm{e}^{-\zeta\,\omega_n\,(t-\mathbb{t}_{\mathrm{L}})}\left[x(\mathbb{t}_{\mathrm{L}})\cos(\omega_D(t-\mathbb{t}_{\mathrm{L}}))+\frac{y(\mathbb{t}_{\mathrm{L}})+\zeta\,\omega_n\,x(\mathbb{t}_{\mathrm{L}})}{\omega_D}\sin(\omega_D(t-\mathbb{t}_{\mathrm{L}}))\right] \\
& +\frac{1}{m\,\omega_D}\int_0^{(t-\mathbb{t}_{\mathrm{L}})}(pp(s)-m\,e?(s))\,\mathrm{e}^{-\zeta\,\omega_n((t-\mathbb{t}_{\mathrm{L}})-s)}\sin(\omega_D((t-\mathbb{t}_{\mathrm{L}})-s))\,ds
\end{aligned}
\tag{5}
$$

$$
\begin{aligned}
y(t-\mathbb{t}_{\mathrm{L}}) \;=\; \mathcal{D}x(t-\mathbb{t}_{\mathrm{L}}) \;=\;& \\
& \mathrm{e}^{-\zeta\,\omega_n(t-\mathbb{t}_{\mathrm{L}})}\left[y(\mathbb{t}_{\mathrm{L}})\cos(\omega_D(t-\mathbb{t}_{\mathrm{L}}))-\frac{\omega_n}{\omega_D}(\zeta\,y(\mathbb{t}_{\mathrm{L}})+\omega_n\,x(\mathbb{t}_{\mathrm{L}}))\sin(\omega_D(t-\mathbb{t}_{\mathrm{L}}))\right] \\
& +\frac{1}{m}\int_0^{(t-\mathbb{t}_{\mathrm{L}})}(pp(s)-m\,e?(s))\times \\
& \qquad \mathrm{e}^{-\zeta\,\omega_n((t-\mathbb{t}_{\mathrm{L}})-s)}\Big(\cos(\omega_D((t-\mathbb{t}_{\mathrm{L}})-s))-\frac{\zeta\,\omega_n}{\omega_D}\sin(\omega_D((t-\mathbb{t}_{\mathrm{L}})-s))\Big)\,ds
\end{aligned}
\tag{6}
$$

The idea now is to tailor the various parameters of the model in such a way that we can prove that the form we choose for *pp* lets us derive the desired bound $|x| \leq X_B$.

To conform to engineering practice for this class of systems, the form we choose for *pp* will consist of pulses, as suggested earlier. Pulses have a large value for a short period, and are zero the rest of the time. As clearly explained in [19], if the support of a pulse is small, its precise shape has little effect on the dynamics, and only its overall impulse (i.e. integral) matters. Thus the natural temptation is to idealise the pulse into a 'delta function', which has zero duration but nonzero integral. Although no engineering equipment implements a delta function pulse, the idealisation simplifies calculations, and so we will pursue it here, since the deviation from a realistic pulse will be small. Technically, the idealisation also allows us to illustrate how delta functions can be handled in HEB.[6]

Tacitly, we can identify the time period T_P in Fig. 7 with the interval between pulses. Suppose then that one of these idealised delta pulses has just occurred. In the ensuing interval, the form of *pp* will be zero, so the *pp* terms can be removed from (5) and (6). Assuming that we know $x(\mathbb{t}_{\mathrm{L}})$ and $y(\mathbb{t}_{\mathrm{L}})$, we thus calculate the behaviour of x and y in the ensuing interval. Demanding that this remains within safe limits imposes constraints on $x(\mathbb{t}_{\mathrm{L}})$ and $y(\mathbb{t}_{\mathrm{L}})$, which it was the obligation of the immediately preceding pulse to have ensured. Analogously, it is the obligation of the next pulse to ensure equally safe conditions for the next interval. And so on.

Thus, we are interested in estimating the behaviour of the x and y variables during a transition of the *MONITOR* event. To this end, we argue as follows. Having tacitly arranged that the pulses occur at the transitions specified by the *MoSkip* event,

[6] The issue is not a trivial one. HEB semantics is defined in terms of piecewise absolutely continuous functions [11]. But a delta function is not piecewise absolutely continuous, because, to be precise, it is not a function at all.

and thus that pp is zero during a *MONITOR* transition, we note that the period of the building's vibrations during an earthquake, which is typically of the order of a second or two and is captured in the constants ω_n and ζ, will be much longer than the response time of the active protection system, i.e. will be much longer than T_P. Therefore, the domain of integration in (5) and (6) will always be much shorter than a half cycle of the trigonometric terms, as a consequence of which the combined exponential and trigonometric terms will always be positive throughout the domain of integration. In such a case, the extremal values of the integral will arise when the modulating factor $e?$ takes its own extremal values. These are just the constant values $e \pm e_B$ (the sign to be chosen depending on which one favours the argument we wish to make). Substituting these (and keeping both signs in case of future need) reduces the integrals to an analytically solvable form, which is readily evaluated [26, 23]. For a duration T_P we get:

$$\begin{aligned}
&x(T_P + \mathbb{t}_\mathrm{L}) \;=\\
&\quad \mathrm{e}^{-\zeta\,\omega_n T_P}\left[x(\mathbb{t}_\mathrm{L})\cos(\omega_D T_P) + \frac{1}{\sqrt{1-\zeta^2}}\Big(\frac{1}{\omega_n}\,y(\mathbb{t}_\mathrm{L}) + \zeta\,x(\mathbb{t}_\mathrm{L})\Big)\sin(\omega_D T_P)\right]\\
&\quad -(e \pm e_B)\,\frac{1}{\omega_n^2}\left[1 - \mathrm{e}^{-\zeta\,\omega_n T_P}\Big(\cos(\omega_D T_P) + \frac{\zeta}{\sqrt{1-\zeta^2}}\sin(\omega_D T_P)\Big)\right]
\end{aligned} \tag{7}$$

$$\begin{aligned}
&y(T_P + \mathbb{t}_\mathrm{L}) \;=\; \mathcal{D}\,x(T_P + \mathbb{t}_\mathrm{L}) \;=\\
&\quad \mathrm{e}^{-\zeta\,\omega_n T_P}\left[y(\mathbb{t}_\mathrm{L})\cos(\omega_D T_P) - \frac{1}{\sqrt{1-\zeta^2}}\Big(\omega_n\,x(\mathbb{t}_\mathrm{L}) + \zeta\,y(\mathbb{t}_\mathrm{L})\Big)\sin(\omega_D T_P)\right]\\
&\quad -(e \pm e_B)\,\frac{1}{\omega_D}\,\mathrm{e}^{-\zeta\,\omega_n T_P}\sin(\omega_D T_P)
\end{aligned} \tag{8}$$

Next we observe that the impulsive force that the active protection system applies during a pulse will not significantly change x but will only have a significant impact on y. Thus, assuming the system only reacts when $|x|$ is close to its permitted maximum value (specified by an appropriately chosen threshold value X_{th}), we infer that the following statements:

$$\textsc{if}\;\; 0 < X_{th} \le x(\mathbb{t}_\mathrm{L}) \le X_B \;\;\textsc{then ensure}\;\; x(T_P + \mathbb{t}_\mathrm{L}) \le X_B \;\;\textsc{fi} \qquad \text{and} \tag{9}$$

$$\textsc{if}\;\; 0 > -X_{th} \ge x(\mathbb{t}_\mathrm{L}) \ge -X_B \;\;\textsc{then ensure}\;\; x(T_P + \mathbb{t}_\mathrm{L}) \ge -X_B \;\;\textsc{fi} \tag{10}$$

express a policy for ensuring that the invariant $|x| \le X_B$ is maintained throughout the dynamics of the system. These allow us to focus predominantly on equation (7), using (8) only occasionally.

We note that for the typical scenario of interest, $\zeta \lesssim 0.1$, so that $\sqrt{1 \pm \zeta^2} \cong 1$. From this we deduce that $\omega_n \cong \omega_D$, so we call both of them ω henceforth. Bearing the implications of such system parameters in mind, we embark on a process of simplifying (7) and (8). The observations just made lead to:

$$\begin{aligned} x(T_P+\mathbb{t}_{\mathrm{L}}) \;=\; & \\ & \mathrm{e}^{-\zeta\,\omega\,T_P}\Big[x(\mathbb{t}_{\mathrm{L}})\cos(\omega\,T_P)+\Big(\frac{1}{\omega}\,y(\mathbb{t}_{\mathrm{L}})+\zeta\,x(\mathbb{t}_{\mathrm{L}})\Big)\sin(\omega\,T_P)\Big] \\ & -(e\pm e_B)\,\frac{1}{\omega^2}\Big[1-\mathrm{e}^{-\zeta\,\omega\,T_P}\Big(\cos(\omega\,T_P)+\zeta\,\sin(\omega\,T_P)\Big)\Big] \end{aligned} \tag{11}$$

$$\begin{aligned} y(T_P+\mathbb{t}_{\mathrm{L}}) \;=\; & \\ & \mathrm{e}^{-\zeta\,\omega\,T_P}\Big[y(\mathbb{t}_{\mathrm{L}})\cos(\omega\,T_P)-\Big(\omega\,x(\mathbb{t}_{\mathrm{L}})+\zeta\,y(\mathbb{t}_{\mathrm{L}})\Big)\sin(\omega\,T_P)\Big] \\ & -(e\pm e_B)\,\frac{1}{\omega}\,\mathrm{e}^{-\zeta\,\omega\,T_P}\sin(\omega\,T_P) \end{aligned} \tag{12}$$

Also, to ensure that the system is responsive enough to adequately dampen large oscillations coming from an earthquake, it should be prepared to respond at least 20 times per building oscillation, making $\omega T_P \cong 0.05$, and making $\zeta\,\omega T_P \cong 0.005$. This allows us to further simplify (11) and (12), keeping low order terms only. We work to second order in ωT_P and regard $\zeta \approx \omega T_P$. This leads to the discarding of contributions $[(1/2)\zeta^2\,\omega^2\,T_P^2]\,(y(\mathbb{t}_{\mathrm{L}})\,T_P)$ to $x(T_P+\mathbb{t}_{\mathrm{L}})$ and of $\zeta^2\,\omega^2\,T_P^2\,y(\mathbb{t}_{\mathrm{L}})+\zeta\,\omega^2\,T_P^2\,(x(\mathbb{t}_{\mathrm{L}})\,\omega)-(e\pm e_B)\,T_P\,((1/2)\,\zeta^2\,\omega^2\,T_P^2)$ to $y(T_P+\mathbb{t}_{\mathrm{L}})$ — these will certainly be negligible if we consider that real systems are noisy. In this way we get:

$$\begin{aligned} x(T_P+\mathbb{t}_{\mathrm{L}}) \;=\; & \\ & \Big[x(\mathbb{t}_{\mathrm{L}})\Big(1-\frac{\omega^2\,T_P^2}{2}\Big)+y(\mathbb{t}_{\mathrm{L}})\,T_P\Big(1-\zeta\,\omega\,T_P\Big)\Big]-\frac{(e\pm e_B)\,T_P^2}{2} \end{aligned} \tag{13}$$

$$\begin{aligned} y(T_P+\mathbb{t}_{\mathrm{L}}) \;=\; & \\ & \Big[y(\mathbb{t}_{\mathrm{L}})\Big(1-2\,\zeta\,\omega\,T_P\Big)-\omega\,x(\mathbb{t}_{\mathrm{L}})\,\omega\,T_P\Big]-(e\pm e_B)\,T_P\Big(1-\zeta\,\omega\,T_P\Big) \end{aligned} \tag{14}$$

These formulae exhibit characteristics that we would expect. Thus, the leading contribution to $x(T_P+\mathbb{t}_{\mathrm{L}})$ is $x(\mathbb{t}_{\mathrm{L}})+y(\mathbb{t}_{\mathrm{L}})\,T_P$, to which are added smaller corrections, while the leading contribution to $y(T_P+\mathbb{t}_{\mathrm{L}})$ is $y(\mathbb{t}_{\mathrm{L}})$ itself, modified by smaller corrections. The relative constancy of the velocity y over an interval T_P confirms that our proposed strategy, of imposing a pulse which discontinuously alters $y(\mathbb{t}_{\mathrm{L}})$, will be the dominant effect on the displacement variable x during the interval. We also see that the earthquake acceleration, which contributes the $(e\pm e_B)$ terms, is not very significant unless it is violent enough to be comparable to the time period T_P or its square.

In principle (13) and (14) give us enough to design the protection system. At the end of each T_P interval, we examine $x(\mathbb{t}_{\mathrm{L}})$ and $y(\mathbb{t}_{\mathrm{L}})$, we calculate $x(T_P+\mathbb{t}_{\mathrm{L}})$ according to (13), and if the answer exceeds X_B, we apply a pulse to change $y(\mathbb{t}_{\mathrm{L}})$ to a new value $\overline{y(\mathbb{t}_{\mathrm{L}})}$ for which a recalculated $\overline{x(T_P+\mathbb{t}_{\mathrm{L}})}$ does not exceed X_B. However, we wish to do a bit better. We would like to identify a safe region, given by a threshold value X_{th}, such that if $x(\mathbb{t}_{\mathrm{L}}) \leq X_{th}$, no further action is needed. To identify

X_{th}, we need an upper bound for $y(\mathbb{t}_{\mathrm{L}})$ so that we can estimate how much ‘help’ the velocity could give to x during an interval. We argue as follows.

We note that starting from a stationary state, neglecting the lower order corrections (including the contribution from $x(\mathbb{t}_{\mathrm{L}})$ whose coefficient is small), and considering the strongest earthquake the system is designed to cope with, E_B, each interval can add at most $E_B T_P$ to y. So after N intervals, y is at most $N E_B T_P$. Turning to x, an interval can similarly add at most $E_B T_P^2/2$ from the last term of (13), and after N intervals, $(1+2\dots N)E_B T_P$ is added from the velocity term, giving a total, after N intervals, of $E_B T_P^2(N^2+2N)/2$. This must not exceed X_B, which leads to:[7]

$$N \approx \left\lfloor \sqrt{2X_B/E_B T_P^2 + 1} \right\rfloor \tag{15}$$

The threshold value X_{th} must be small enough that the largest possible single increment of x cannot exceed $X_B - X_{th}$. From (13), using the maximal velocity derived earlier, we get:

$$X_{th} \le X_B - \left(E_B T_P^2 \sqrt{2X_B/E_B T_P^2 + 1} + \frac{E_B T_P^2}{2} \right) \tag{16}$$

For (16) to be reasonable, its RHS must be positive, which leads to the consistency condition $2X_B \ge E_B T_P^2$. This is sensible, since if not (and referring to (13)), a single cold start interval could overreach X_B, and the threshold idea would not make sense. (The same condition is also necessary for (15) to yield a positive integer, when the discarded -1 is reinstated.)

From the account above, it is clear that if $|x(\mathbb{t}_{\mathrm{L}})| \le X_{th}$ at the start of an interval, then the system need do nothing. This will be the case most of the time in reality, since the only vibrations sensed will be from normal everyday activity in the building and its surroundings. However, if $|x(\mathbb{t}_{\mathrm{L}})| > X_{th}$, then the more detailed calculation in (13) will be needed, in case there is a risk of exceeding the bound X_B.

These observations underpin the next model in our HEB development, whose transition diagram is in Fig. 8, and the text of which is in Fig. 9. In this model, the *MONITOR* event no longer has a COMPLY clause stipulating the behaviour of the system via an implicitly chosen pp function. In accordance with our discussion, the externally imposed force p is zero during *MONITOR*. The job of ensuring that the invariant $|x| \le X_B$ is maintained becomes the responsibility of delta pulses that jolt the system into acceptable behaviour when necessary.

Here we hit a technical snag, in that delta functions do not exist in the semantics of HEB (see footnote 6). Rather than ex-

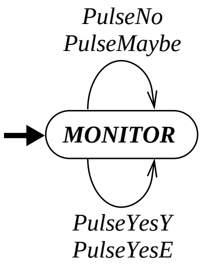


Fig. 8 A transition diagram for the second refinement of the HEB earthquake damage prevention model.

[7] In deriving this, we dropped a term -1 from the RHS of (15).

```
MACHINE $ActConMch\_2$
REFINES $ActConMch\_1$
CLOCK $clk\_pls$
PLIANT $x, y, p$
INVARIANTS
  $x, y, p \in \mathbb{R}, \mathbb{R}, \mathbb{R}$
  $|x| \le X_B$
EVENTS
  $INITIALISATION$
    REFINES $INITIALISATION$
    STATUS ordinary
    BEGIN
      $clk\_pls := 0$
      $x, y, p := 0, 0, 0$
    END
  $PulseNo$
    REFINES $MoSkip$
    STATUS ordinary
    WHEN
      $clk\_pls = T_P \wedge |x| < X_{th}$
    THEN
      $clk\_pls := 0$
    END
  $PulseMaybe$
    REFINES $MoSkip$
    STATUS ordinary
    ANY $e?$
    WHERE
      $clk\_pls = T_P \wedge |x| \ge X_{th} \wedge$
      $e? \in \mathbb{R} \wedge |e?| \le E_B \wedge$
      $\Big| x\big(1 - \omega^2 T_P^2/2\big) + y T_P\big(1 - \zeta\,\omega T_P\big)$
        $- e? T_P^2/2 \Big| \le X_B$
    THEN
      $clk\_pls := 0$
    END
  $MONITOR$
    REFINES $MONITOR$
    STATUS pliant
    ANY $e, e?$
    WHERE
      $e \in \mathbb{R} \wedge \mathsf{CONST}(e) \wedge$
      $e? \in \mathbb{R} \wedge |e?| \le E_B \wedge |e - e?| \le e_B$
    WITH $pp = 0$
    SOLVE
      $p := 0$
      $\mathcal{D}x = y$
      $\mathcal{D}y = -\frac{c}{m}y - \frac{k}{m}x - e?$
    END
...  ...
```

```
...  ...
  $PulseYesY$
    REFINES $MoSkip$
    STATUS ordinary
    ANY $\Delta x, w, e?$
    WHERE
      $clk\_pls = T_P \wedge |x| \ge X_{th} \wedge$
      $e? \in \mathbb{R} \wedge |e?| \le E_B \wedge$
      $\Big| x\big(1 - \omega^2 T_P^2/2\big) + y T_P\big(1 - \zeta\,\omega T_P\big)$
        $- e? T_P^2/2 \Big| - X_B = w \wedge$
      $w > 0 \wedge$
      $\Delta x = w + (X_B - |x|) \wedge$
      $\big[(\mathrm{sign}(y) = \mathrm{sign}(-e?) \wedge$
          $|y T_P(1 - \zeta\,\omega T_P)| \ge \Delta x/2) \vee$
       $(\mathrm{sign}(y) \ne \mathrm{sign}(-e?) \wedge$
          $|y T_P(1 - \zeta\,\omega T_P)| \ge \Delta x)\big]$
    THEN
      $clk\_pls := 0$
      $y := -y$
    END
  $PulseYesE$
    REFINES $MoSkip$
    STATUS ordinary
    ANY $\Delta x, w, e?$
    WHERE
      $clk\_pls = T_P \wedge |x| \ge X_{th} \wedge$
      $e? \in \mathbb{R} \wedge |e?| \le E_B \wedge$
      $\Big| x\big(1 - \omega^2 T_P^2/2\big) + y T_P\big(1 - \zeta\,\omega T_P\big)$
        $- e? T_P^2/2 \Big| - X_B = w \wedge$
      $w > 0 \wedge$
      $\Delta x = w + (X_B - |x|) \wedge$
      $\big[(\mathrm{sign}(y) = \mathrm{sign}(-e?) \wedge$
          $|- e? T_P^2/2| \ge \Delta x/2) \vee$
       $(\mathrm{sign}(y) \ne \mathrm{sign}(-e?) \wedge$
          $|- e? T_P^2/2| \ge \Delta x)\big]$
    THEN
      $clk\_pls := 0$
      $y := e? T_P/2\,(1 - \zeta\,\omega T_P)$
    END
END
```

Fig. 9 Second refinement of the system.

press the needed delta functions directly, we use their time integrals, which are discontinuous functions, which *do* exist in the semantics of HEB, and are typically implemented using mode events. The burden of implementing the pulses thus falls to refinements of the earlier *MoSkip* event, which implement the imposition of the needed delta functions onto the acceleration $\mathcal{D}\mathcal{D}x = \mathcal{D}y$, by instead imposing discontinuities on its integral y.

Accordingly, when time is a multiple of T_P, if $|x| < X_{th}$, then event *PulseNo* executes, and just resets the clock. But if $|x| \geq X_{th}$ then we need a more complex calculation, analogous to equation (13). If this reveals that the projected future $|x|$ value will nevertheless still be below X_B, then the action is the same, expressed in event *PulseMaybe*.

However, if the calculation reveals that without intervention X_B will be breached, then the system must intervene to prevent it. This is captured in mode events *PulseYesY* and *PulseYesE* and involves a case analysis as follows.

Let us call Δx the difference between the projected future $|x|$ value and the before-value of $|x|$ in these events, as in the two events' guards. Then if Δx turns out positive, it can only be because either the y term or the $-e?$ term of the projected future $|x|$ value, or both, is/are driving $|x|$ too high. At least one of these terms has a value whose sign is the same as that of the before-value of x in the two events, else both terms would drive $|x|$ smaller, contradicting the breaching of X_B. N. B. We assume that the threshold is big enough that above threshold, a single interval cannot cause x to change sign, and thus cannot cause $|x|$ to increase even in cases in which the rate of change of x changes sign.

Suppose then that both terms have values whose sign agrees with that of the value of x. Then one of them has a value which is at least $\Delta x/2$ since they act additively and their sum is Δx. In this case it is sufficient to invert the sign of the larger contribution to ensure that their net effect diminishes $|x|$. So we either flip y, or flip a suitably rescaled $e?$. This covers one of the two cases in each of *PulseYesY* and *PulseYesE*.

Suppose alternatively that only one of the terms has a value whose sign agrees with that of the value of x. Then the magnitude of that term must exceed Δx, since they act subtractively and the difference of their magnitudes is still Δx.[8] In this case it is sufficient to invert the sign of this larger contribution to ensure their net effect diminishes $|x|$. This covers the remaining two cases in *PulseYesY* and *PulseYesE*.

8 On HEB Refinement

At this point we reflect on the refinement just done. A first point notes that during normal Event-B refinement [3], the behaviour of an event is typically restricted, making it more deterministic. In our case, we have taken this to an extreme, by ef-

[8] Mathematically, it is possible for both terms to have magnitude bigger than Δx, unless we take into account relevant upper bounds etc. and show that it is impossible. We will just assume that there is no such possibility in our problem space.

fectively abandoning external control of the behaviour of the dynamical variables x and y via p during *MONITOR*, and have delegated this duty instead to the *PulseXX* events. So the *PulseXX* events are new in the model of Fig. 9, and define new behaviour for y (and potentially for x too, if it were needed). This is against the rules of Event-B refinement, since new behaviour for variables should be introduced at the same time as the variables themselves (being made more deterministic subsequently) — whereas we introduced x and y during the previous refinement.

We partly mitigated this by making the *PulseXX* events refine the earlier *MoSkip* events, introduced during the previous refinement stage, and giving the *MoSkip* events the status 'anticipating'. This status allows an event, newly introduced during a refinement step, and which would normally be required to strictly decrease a relevant *variant* function (to ensure the convergence of the new behaviour), to *not strictly* decrease the variant then, postponing this obligation till later.[9] Since we included no variants in our development, we discharged this duty trivially. The introduction of *MoSkip* and variable y at the same time thus not only allowed fresh choice of e in successive interations of *MONITOR* but allowed the manipulation of y in refinements of *MoSkip*.

Unfortunately though, by not mentioning y at all, *MoSkip* by default specifies that y does not change during *MoSkip* transitions, while the *PulseYes* events refining it specify nontrivial changes to y. It is tempting to think that this is still a refinement, since the only invariant concerning y is $y \in \mathbb{R}$, the weakest possible invariant. However, when the same variable exists in a machine and in its refinement, there is an implicit equality invariant between the abstract and concrete versions of the variable — otherwise writing more conventional refinements would become intolerably verbose. In this regard, 'no change' in *MoSkip* is incompatible with 'nontrivial update' in *PulseYes*, and our refinement isn't quite legal after all. This shows that 'refining to a delta function' is not ideally handled in HEB. The only way to make the development unimpeachable according to the rules of Event-B is to introduce the variable y and the nontrivial mode event behaviour at the same time. But this is less desirable from our point of view, as it forces the choice of control strategy without permitting consideration of alternatives, and flies in the face of the objectives of a refinement driven development strategy which aims at introducing detail into designs in stages.

A second point concerns the arguments we employed in the preceding pages. Our reasoning started out being quite watertight mathematically, but rather quickly, we started to introduce simplifications which were perfectly justifiable on engineering grounds, but which would not pass the unblinking scrutiny of formal proof. Two centuries or more of rigorous mathematical analysis have, in principle, developed techniques, using which, such a shortcoming could be overcome, but the amount of work involved would be considerable, and would quickly surpass the small amount of added assurance that could be gained. The formal development field, in its somewhat strenuous avoidance of engagement with continuous mathematics hitherto, has not really developed a cost effective approach to dealing with this issue.

[9] The formal presentation of HEB [11] does not mention the anticipating status, since that is somewhat outside the main concerns there. But there is no reason to forbid it since it concerns purely structural matters.

A third point concerns the extent to which the model of Fig. 9 can actually be proved correct using a *per event* proving strategy as embodied in the semantics and verification architecture of Event-B. This is by contrast with the arguments in preceding pages, which focused on the application structure and employed whatever observations seemed useful at the time, without regard to how the various points were to be structured into an overall correctness argument. Here, the news, though not perfect, is better.

We note that the *MONITOR* event, as written in Fig. 9, cannot by itself be correct according to the normal 'preserving the invariant' notion of event correctness, since it demands no restrictions on $x(\mathbb{t}_{\mathrm{L}})$ and $y(\mathbb{t}_{\mathrm{L}})$. Without prior knowledge about these, the ODE system can easily breach the $x \leq X_B$ bound during a T_P interval. Of course, we rely on the *PulseYes* events to ensure appropriate $x(\mathbb{t}_{\mathrm{L}})$ and $y(\mathbb{t}_{\mathrm{L}})$ values for the subsequent *MONITOR* event, but the *MONITOR* event correctness proof obligations know nothing of this. However, in HEB, we also have 'well-formedness' proof obligations, that police the handover between mode and pliant events. These can check that after any of the *PulseXX* events, the values of x and y are appropriate. In particular, they check that after the *PulseXX* events the guard for at least one pliant event is true. Since we have designed the *PulseXX* events to ensure exactly what is required here, the trivial guard of the *MONITOR* event of Fig. 9 could, in fact, be strengthened to demand a suitably stringent constraint on $x(\mathbb{t}_{\mathrm{L}})$ and $y(\mathbb{t}_{\mathrm{L}})$, from which, 'preserving the invariant' would become possible. So, although we did not get diverted by this detail earlier, a solution entirely within the rules is available.

9 Sensors, Actuators, Sampling, Quantization, Decomposition

The next model in our development tackles a number of issues that add low level complexity. Following the structure of [20, 21], we introduce a sensor and an actuator into the system. Elements like these bring various kinds of imprecision to the development. Thus, they typically act at discrete moments of time — this brings temporal imprecision. Their inputs and outputs typically have finite ranges, and are quantized — this brings imprecision of magnitude. The impact of these sources of imprecision is similar from a formal point of view, and describing these phenomena precisely, generates complexity in the textual description of the system.

Moreover, a model close to the architectural structure of [20, 21] would place the architecturally distinct components of the system in separate constructs. To create such a model requires the decomposition of a monolithic version into smaller pieces, a process which, if done with precision, generates both textual complexity and a lot of repetition of the model text.

To minimise verbosity, our strategy will therefore be as follows. Viewing the model of Fig. 9 as being at level 2, the level 2 model is conceptually developed into an unstated, but still monolithic model, incorporating the features mentioned above, at level 3 (with machine *ActConMch_3* say). This is then decomposed into a multimachine project at level 4, exhibiting the desired architectural structure. The

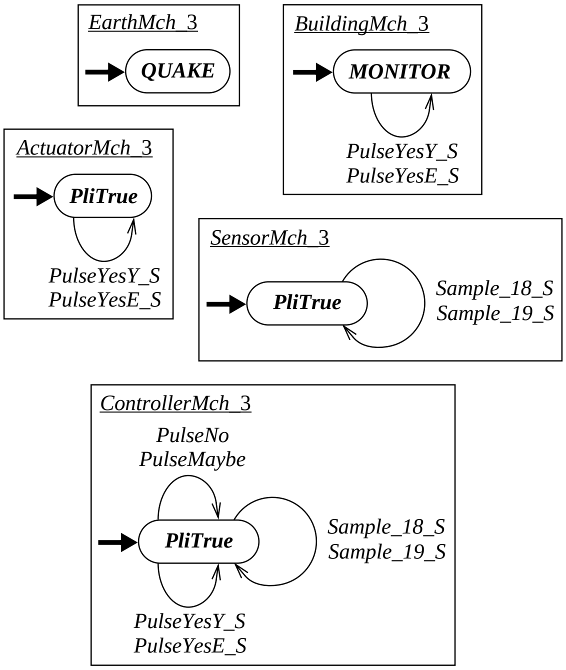


Fig. 10 A family of transition diagrams for the HEB machines of a distributed concurrent version of the active earthquake damage prevention system.

level 4 model is presented in Figs. 11-14, and described below. We comment more extensively on the level 4 model later on.

In Fig. 10 we have a depiction of the various HEB machines of the distributed concurrent HEB model that results from the process just sketched. Figs. 11-14 contain the text of the resulting model. We start with the PROJECT *ActCon_4_Prj* file in Fig. 11, which describes the overall structure. The DECOMPOSES *ActCon_3_Prj* line refers to the fictitious level 3 system, of which more later. The main job of the PROJECT file is to name the constituent machines and interfaces, and to define needed synchronisations between the mode events of the different machines. Thus there are machines for the earth, the building, the actuator, the sensor, and the controller. The PROJECT file also names the INTERFACE *ActCon_4_IF* file. This declares any variables that are shared between more than one machine, their initialisations, and, most importantly, any invariants that mention any of these variables. (The latter point can place stringent restrictions on how variables are partitioned into different interfaces and machines.) The final responsibility of the PROJECT file is to declare the mode event synchronisations. Thus SYNCHronisation *Sample*18 specifies that mode event *Sample_*18*_S* in machine *SensorMch_*4 and mode event *Sample_*18*_S* in machine *ControllerMch_*4 must be executed simultaneously. This means that they can only execute if all the guard conditions in all the events of

```
PROJECT ActCon_4_Prj
  DECOMPOSES ActCon_3_Prj
  MACHINE EarthMch_4
  MACHINE BuildingMch_4
  MACHINE SensorMch_4
  MACHINE ControllerMch_4
  MACHINE ActuatorMch_4
  INTERFACE ActCon_4_IF
  SYNCH(Sample18)
    SensorMch_4.Sample_18_S
    ControllerMch_4.Sample_18_S
  END
  SYNCH(Sample19)
    SensorMch_4.Sample_19_S
    ControllerMch_4.Sample_19_S
  END
  SYNCH(PulseYesY)
    ActuatorMch_4.PulseYesY_S
    BuildingMch_4.PulseYesY_S
    ControllerMch_4.PulseYesY_S
  END
  SYNCH(PulseYesE)
    ActuatorMch_4.PulseYesE_S
    BuildingMch_4.PulseYesE_S
    ControllerMch_4.PulseYesE_S
  END
END
```

```
INTERFACE ActCon_4_IF
PLIANT xx, yy, ee
INVARIANTS
  $xx, yy, ee \in \mathbb{R}, \mathbb{R}, \mathbb{R}$
  $|xx| \leq X_B$
INITIALISATION
  $xx, yy, ee := 0, 0, 0$
END
```

Fig. 11 The PROJECT and INTERACE files of the further developed and decomposed system.

the synchronisation are true. The same remarks apply to the other synchronisations declared in the project file.

We turn to the machines, pictured in Fig. 10. In outline, machine *EarthMch_4* is responsible for producing the earthquake acceleration, which comes from the input *e*?, as in previous models. This is simply captured during the pliant event *QUAKE* and is recorded in the shared pliant variable *ee*, declared in the interface (which the *EarthMch_4* machine CONNECTS to). We can see that the *QUAKE* event comes from decomposing the earlier *MONITOR* pliant event, and we will see the remnants of the *MONITOR* event elsewhere soon. Since any HEB machine must have at least one pliant event to describe what happens over the course of time, but need not contain any other event, and since *QUAKE* addresses that requirement, there are no other events in *EarthMch_4*.

The shared variable *ee* is accessed by machine *BuildingMch_4*. This contains the remainder of the earlier *MONITOR* pliant event, namely the ODE system defining the building's response, which uses *ee*. It also contains the business end of the *PulseYesY_S* and *PulseYesE_S* mode events, which take their inputs (which are received from the actuator using input *ys*?) and discontinuously impose the received values on the velocity variable *yy*.

```
MACHINE EarthMch_4
CONNECTS ActCon_4_IF
  QUAKE
    STATUS pliant
    ANY $e, e?$
    WHERE
      $e \in \mathbb{R} \wedge \mathsf{CONST}(e) \wedge$
      $e? \in \mathbb{R} \wedge |e?| \leq E_B \wedge |e - e?| \leq e_B$
    BEGIN
      $ee := e?$
    END
END
```

```
MACHINE BuildingMch_4
CONNECTS ActCon_4_IF
EVENTS
  PulseYesY_S
    STATUS ordinary
    ANY $ys?$
    WHERE
      $ys? \in \mathbb{R}$
    THEN
      $yy := ys?$
    END
  PulseYesE_S
    STATUS ordinary
    ANY $ys?$
    WHERE
      $ys? \in \mathbb{R}$
    THEN
      $yy := ys?$
    END
  MONITOR
    STATUS pliant
    SOLVE
      $\mathcal{D} xx = yy$
      $\mathcal{D} yy = -\frac{c}{m} yy - \frac{k}{m} xx - ee$
    END
END
```

```
MACHINE SensorMch_4
CONNECTS ActCon_4_IF
EVENTS
  Sample_18_S
    ANY $sens\_x!$
    WHERE
      $sens\_x! \in \mathbb{R}$
    THEN
      $sens\_x! := K_{xsqs}^{-1} \lfloor K_{xsqs}\, xx \rceil$
    END
  Sample_19_S
    ANY $sens\_x!, sens\_e!$
    WHERE
      $sens\_x! \in \mathbb{R} \wedge sens\_e! \in \mathbb{R}$
    THEN
      $sens\_x! := K_{xsqs}^{-1} \lfloor K_{xsqs}\, xx \rceil$
      $sens\_e! := K_{esqs}^{-1} \lfloor K_{esqs}\, ee \rceil$
    END
  PliTrue
    STATUS pliant
    COMPLY INVARIANTS
    END
END
```

```
MACHINE ActuatorMch_4
EVENTS
  PulseYesY_S
    ANY $ys!, act\_y?$
    WHERE
      $ys! \in \mathbb{R} \wedge act\_y? \in \mathbb{R}$
    THEN
      $ys! := K_{ysqs}^{-1} \lfloor K_{ysqs}\, act\_y? \rceil$
    END
  PulseYesE_S
    ANY $ys!, act\_y?$
    WHERE
      $ys! \in \mathbb{R} \wedge act\_y? \in \mathbb{R}$
    THEN
      $ys! := K_{ysqs}^{-1} \lfloor K_{ysqs}\, act\_y? \rceil$
    END
  PliTrue
    STATUS pliant
    COMPLY INVARIANTS
    END
END
```

Fig. 12 Machines for earth, building, sensor and actuator.

We come to the *SensorMch_4* and *ActuatorMch_4* machines. Their behaviour is essentially discrete, so to satisfy the requirement for having a pliant event, both machines have a default COMPLY *INVARIANTS* pliant event, named, as is typically the case, *PliTrue*. In fact, since all the pliant variables are handled by other machines, there is nothing for these *PliTrue* events to do, and that is part of the semantics of 'COMPLY *INVARIANTS*' in HEB.

The job of the *SensorMch_4* machine is to sample the physical values required by a deidealised implementation of the system. The values are required at pulse issuing time, but to allow time for computation, as in [20, 21], they are collected a little earlier. The position and earth acceleration values, from *xx* and *ee*, are collected $19/20$ of the way through a T_P interval, and are transmitted (to the controller machine) in output variable *sens_x*! and *sens_e*!. An extra position value is needed for calculating a velocity estimate, so another sample of *xx* is taken $18/20$ of the way through T_P. Notice that the *xx* and *ee* values are scaled (by K_{xsqs} and K_{esqs}), rounded, and then unscaled (by K_{xsqs}^{-1} and K_{esqs}^{-1}) before sending, to model the quantization process.[10] The mode events that do these jobs are *Sample_18_S* and *Sample_19_S*. The '*_S*' suffixes on these names indicate, for readability, that these are synchronised with mode events in one or more other machines, though, as we mentioned earlier, the formal definition of a project's synchronisations are in the project file.

The same general comments work for the *ActuatorMch_4* machine. Only the velocity variable is modified in our development, so only this variable is acted on by the actuator. The value needed is received (from the controller machine) in the *act_y*? input of synchronised events *PulseYesY_S* and *PulseYesE_S*, and after quantization via K_{ysqs} and its inverse, is transmitted (to the building machine) in the *ys*! output.[11] As for the sensor, there is no need for any non-trivial pliant event, so a default *PliTrue* suffices.

At the heart of the system is the *ControllerMch_4* machine. This houses the remaining functionality, and the non-trivial computation. The clock *clk_pls* is declared here, as are local variables $x18, x19, y19, e19$, the sampled values of the dynamical variables, which are not needed in any other machine. We see also that *ControllerMch_4* only requires the *PliTrue* default pliant event, since its interventions are exclusively at individual moments of time. It also contains the remaining portions of the various synchronisations we have discussed.

The *Sample_18_S* event picks up the sampled position at times $18/20\,T_P$ of an interval, recording them in $x18$. The *Sample_19_S* event picks up position and acceleration samples at $19/20\,T_P$ and, as well as recording these, it calculates an estimate of velocity from the position samples and records it in $y19$. The values in $x19, y19, e19$ are then ready for the pulse calculations, which would consume some time to do, but which are modelled as taking place instantaneously at the end of the interval in the various *Pulse* events.

Compared with the other events, events *Sample_18_S* and *Sample_19_S* are newly introduced in this development step. In such a case, Event-B practice asks that they strictly decreases some variant function, which is included in Fig. 13 after the in-

[10] In reality, a sensor would send values in its own units, and scaling would be done as part of the controller's job, but we avoid this so as to keep the controller calculation reasonably transparent.

[11] On a technical level, the building and actuator machines illustrate the pattern whereby synchronised mode events in different machines can instantaneously share values: one event uses an output variable and the others use an input variable with a complementary (CSP style) name.

```
MACHINE $ControllerMch\_4$
CONNECTS $ActCon\_4\_IF$
CLOCK $clk\_pls$
VARIABLES
  $x18, x19, y19, e19$
INVARIANTS
  $x18, x19, y19, e19 \in \mathbb{R}, \mathbb{R}, \mathbb{R}, \mathbb{R}$
VARIANT $\lceil (T_P - clk\_pls) \times 20 \rceil$
EVENTS
  $INITIALISATION$
    STATUS ordinary
    BEGIN
      $clk\_pls := 0$
      $x19, y19, e19 := 0, 0, 0$
    END
  $PliTrue$
    STATUS pliant
    COMPLY $INVARIANTS$
    END
  $Sample\_18\_S$
    ANY $sens\_x?$
    WHEN
      $clk\_pls = \frac{18}{20} T_P$
    THEN
      $x18 := sens\_x?$
    END
...  ...
```

```
...  ...
  $Sample\_19\_S$
    ANY $sens\_x?, sens\_e?$
    WHEN
      $clk\_pls = \frac{19}{20} T_P$
    THEN
      $x19 := sens\_x?$
      $y19 := \lfloor (sens\_x? - x18) \frac{20}{T_P} \rceil$
      $e19 := sens\_e?$
    END
  $PulseNo$
    STATUS ordinary
    WHEN
      $clk\_pls = T_P \wedge |x19| < X_{thsq}$
    THEN
      $clk\_pls := 0$
    END
  $PulseMaybe$
    STATUS ordinary
    WHEN
      $clk\_pls = T_P \wedge |x19| \geq X_{thsq} \wedge$
 $\Big| x19 \big(1 - \omega^2 T_P^2/2\big) + y19\, T_P \big(1 - \zeta\, \omega\, T_P\big)$
          $- e19\, T_P^2/2 \Big| \leq X_{Bsq}$
    THEN
      $clk\_pls := 0$
    END
...  ...
```

Fig. 13 The controller machine, first part.

variants. It is clear that at the two occurrences of the *Sample* events in each interval, the value of the variant drops, firstly from $\lim_{\varepsilon \to 0+} \lceil (T_P - (\frac{18}{20} T_P + \varepsilon)) \times 20 \rceil = 3$ to 2, and then from 2 to 1,[12] thus strictly decreasing it, as required.

The *PulseNo*, *PulseMaybe*, and now synchronised *PulseYesY_S* and *PulseYesE_S* events, handle the needed responses to building movement, as before, except that the calculations are now done using the sampled, quantized (SQ) values rather than the ideal, instantaneous (II) ones. This inevitably leads to disagreement with the ideal calculations in the border country where different behaviour regimes meet (in our case, the border country between the do pulse and don't pulse regimes).[13]

In our case, we have to cope with the possibility that the SQ values dictate a pulse in a situation where the II values don't (which is tolerable, since it will only happen in the border country, where pulses are probable anyway), or that the SQ values don't dictate a pulse in a situation where the II values do (which is *intolerable* since

[12] Note that this critically depends on insisting that intervals of pliant behaviour are left closed and right open.

[13] Henceforth, we will use SQ to refer to and to label elements and quantities relevant to the level 4 model of Figs. 11-14 (and, implicitly to its unstated level 3 precursor), and II for elements relevant to the level 2 model of Fig. 9, as needed.

```
...   ...
    $PulseYesY\_S$
      STATUS  ordinary
      ANY  $\Delta x_{sq}, w_{sq}$
      WHERE
        $clk\_pls = T_P \wedge |x19| \geq X_{thsq} \wedge$
  $\Big| x19\,(1-\omega^2 T_P^2/2) + y19\,T_P(1-\zeta\,\omega T_P)$
            $-\,e19\,T_P^2/2\,\Big| - X_{Bsq} = w_{sq} \wedge$
        $w_{sq} > 0 \wedge$
        $\Delta x_{sq} = w_{sq} + (X_{Bsq} - |x19|) \wedge$
        $\big[(\mathrm{sign}(y19) = \mathrm{sign}(-e19) \wedge$
            $|y19\,T_P(1-\zeta\,\omega T_P)| \geq \Delta x_{sq}/2) \vee$
          $(\mathrm{sign}(y19) \neq \mathrm{sign}(-e19) \wedge$
            $|y19\,T_P(1-\zeta\,\omega T_P)| \geq \Delta x_{sq})\big]$
      THEN
        $clk\_pls$ := $0$
        $act\_y!$ := $-y19$
      END
...   ...
```

```
...   ...
    $PulseYesE\_S$
      STATUS  ordinary
      ANY  $\Delta x_{sq}, w_{sq}$
      WHERE
        $clk\_pls = T_P \wedge |x| \geq X_{thsq} \wedge$
  $\Big| x19\,(1-\omega^2 T_P^2/2) + y19\,T_P(1-\zeta\,\omega T_P)$
            $-\,e19\,T_P^2/2\,\Big| - X_{Bsq} = w_{sq} \wedge$
        $w_{sq} > 0 \wedge$
        $\Delta x_{sq} = w_{sq} + (X_{Bsq} - |x19|) \wedge$
        $\big[(\mathrm{sign}(y19) = \mathrm{sign}(-e19) \wedge$
            $|-\,e19\,T_P^2/2| \geq \Delta x_{sq}/2) \vee$
          $(\mathrm{sign}(y19) \neq \mathrm{sign}(-e19) \wedge$
            $|-\,e19\,T_P^2/2| \geq \Delta x_{sq})\big]$
      THEN
        $clk\_pls$ := $0$
        $act\_y!$ := $e19\,T_P/2\,(1-\zeta\,\omega T_P)$
      END
  END
```

Fig. 14 The controller machine, second part.

it may permit the physical system to overshoot the X_B bound without the SQ model being aware of it). We must prevent the latter.

The approach we take is to conservatively adjust the constants X_{th}, X_B in the model to new values X_{thsq}, X_{Bsq} that preclude the intolerable omissions at the price of admitting more superfluous pulses.[14] For more convenient discussion, we also renamed the local variables $\Delta x, w$ in Fig. 14 by adding a subscript.

Our remarks indicate that whenever $\mathit{PulseNo}^{\mathrm{SQ}}$ or $\mathit{PulseMaybe}^{\mathrm{SQ}}$ can run, then we must be sure that $\mathit{PulseNo}^{\mathrm{II}}$ or $\mathit{PulseMaybe}^{\mathrm{II}}$ will also run. This implies a condition on their guards.

We take the events individually, starting with $\mathit{PulseNo}^{\mathrm{SQ}}$ and $\mathit{PulseNo}^{\mathrm{II}}$. It is clear that the latter is enabled whenever the former is, provided that $|x19| < X_{thsq} \Rightarrow |xx| < X_{th}$ holds. Of course, $x19$ and xx refer to values at different times, but recalling that X_{th} was derived by estimating the maximum achievable displacement over a whole interval in (16), one twentieth of the same argument will cover the difference between $x19$ and xx. So our implication will hold, provided:

$$X_{thsq} \leq X_{th} - \Big(E_B\,T_P^2\sqrt{2X_B/E_B\,T_P^2+1} + \frac{E_B\,T_P^2}{2}\Big)\Big/20 \tag{17}$$

We see that this is a small correction to X_{th}, which, for typical parameter values, will be negligible in practice, if not in mathematics, confirming the conjecture in footnote 14.

[14] Speaking realistically, in a genuine earthquake scenario, noise and experimental uncertainty are likely to be such that the differences between the ideal and conservative values of the constants vanish into insignificance. But it is worth checking that the mathematics confirms this.

Turning to $\mathit{PulseMaybe}^{\mathrm{SQ}}$ and $\mathit{PulseMaybe}^{\mathrm{II}}$ a similar argument applies. Looking at the relevant guards, we see that as well as (17), we will be able to maintain the invariant $|xx| < X_B$ provided we make an analogous correction to X_B for the purpose of the estimates made in the $\mathit{PulseMaybe}^{\mathrm{SQ}}$ guard:

$$X_{Bsq} \leq X_B - \Big(E_B T_P^2 \sqrt{2X_B/E_B T_P^2 + 1} + \frac{E_B T_P^2}{2}\Big)\Big/20 \qquad (18)$$

With these two cases understood, we see that the final two events are covered also. Both $\mathit{PulseYesY\_S}^{\mathrm{SQ}}$ and $\mathit{PulseYesE\_S}^{\mathrm{SQ}}$ flip the sign of the greatest contribution to the estimated increment in displacement, based on the same estimate made in $\mathit{PulseMaybe}^{\mathrm{SQ}}$.

10 Refinement, Retrenchment and Other Technical Issues

In Figs. 11-14 the only structural directives are DECOMPOSES *ActCon_3_Prj* in the project file, and the CONNECTS *ActCon_4_IF* in the various machine files. There are a number of reasons for this. Firstly, we are presuming that the hard work of refining *ActConMch_2* to incorporate the sensor, actuator and discretization features will have been achieved in the (unstated) *ActConMch_3* machine, the only element of the (unstated) level 3 project *ActCon_3_Prj*.[15] This understood, the job of decomposing a monolithic *ActConMch_3* machine into the components seen in Figs. 11-14 is properly covered by the cited directives. Before continuing, we briefly comment on this by envisaging how Figs. 11-14 might be reassembled into a single construct.

Let us start with the two *Sample_18_S* events, shared between *SensorMch_4* and *ControllerMch_4*, and executed synchronously. In a monolithic *ActConMch_3* (which would take on the duties of both machines), there would be a single *Sample_18* event, with guard $clk\_pls = \frac{18}{20}T_P$ and action $x18 := K_{xsqs}^{-1} \lfloor K_{xsqs}\, xx \rceil$. There is no communication, since all the variables are accessible to the one machine. *Sample_19* follows a similar pattern, with two variables assigned. Thus is the sensor machine's functionality absorbed into one encompassing machine.

The actuator is dealt with similarly, except that the building is involved; the functionality of the building is also absorbed into the single encompassing machine, rather as was the case in the level 2 and earlier models. The earth machine is similarly absorbed, eliminating the need for the shared variable *ee*. This account illustrates, in reverse, how the distributed model of Figs. 11-14 is arrived at, presuming the preexistence of the monolithic version. Note that it is a deliberate design objective of the multimachine HEB formalism that the monolithic and distributed versions should be, in all important aspects, semantically indistinguishable; see [12].

Thus, the *ActConMch_3* is relatively easily imagined, avoiding some verbosity. Less easy is its relationship to the level 2 machine *ActConMch_2* — the discussion

[15] We can also regard all the previous models as each being in its own single machine project.

in Section 8, on implicit equality invariants, flags up that the introduction of imprecision via sampling and quantization may not be unproblematic regarding refinement methodology.

The immediate problem was avoided by renaming variables x,y to xx,yy in the level 4 model. But this raises the question of what the relationship between x,y and xx,yy ought to be. It is a truism in formal development that the stronger the invariants you write, the harder the work to prove that they are maintained, but the stronger the assurance that is gained thereby. And conversely. We might thus ease our task by omitting completely any non-trivial relationship between x,y and xx,yy. But this will not do since we still have the level 2 invariant $x \leq X_B$ to establish, which is rendered impossible in the level 4 model without some coupling between x,y and xx,yy.

The obvious relationship to consider is some sort of accuracy bound relating x and xx, and y and yy. Since the damping factor ζ is positive, the dynamics is asymptotically stable, so we can expect the dynamics to be contracting[16] (although a refinement relationship based on this still often requires appropriate conditions on the constants of the system [35]). To see the contracting nature of our dynamics we first need to rewrite (13) and (14) in terms of dimensionally comparable quantities, for example, in terms of $\tilde{x} \equiv x$ and $\tilde{y} \equiv y/\omega$. When this is done, (13) and (14), viewed as a matrix operating on differences in pairs of values of $\tilde{x},\tilde{y}$, has entries $(\delta_{ij} + (-1)^{[i \geq j]} \varepsilon_{ij})$, where δ_{ij} is the identity, and the ε_{ij} are small and positive, from which the contracting nature of the transformation can be inferred.

With this, we can claim that a single execution of *MONITOR* in each of the II and SQ systems will maintain a joint invariant of the form $||(x,y) - (xx,yy)||_{\tilde{1}} \leq A$,[17] provided it is true at the start, but it does not tell us what value we would need to choose for A for this to be true non-trivially.

The latter problem would require a global analysis which could be quite challenging. The issue is made the more difficult by the possibility mentioned before, whereby imprecision caused by conservative design in the SQ system causes the SQ system to express a pulse whereas the II system does not. If this happened, the $||.||_{\tilde{1}}$ distance between the II and SQ systems would suddenly increase dramatically, even if it was well behaved previously, and it would consequently cause the $||.||_{\tilde{1}}$ norm to function poorly as a joint invariant between II and SQ systems, posing a significant impediment to refinement as an convincing notion for relating the II and SQ systems.

A weakening of the highly demanding refinement concept is the idea of retrenchment [14, 13, 30]. In retrenchment the demand to preserve a ‘nearness’ invariant is relaxed by permitting different conditions to be specified for the before-state and after-state of a pair of transitions in the two systems being compared, and allows constants, such as A, to be declared locally per transition instance, rather than globally, as in a refinement relation. This formulation also permits the two systems to part company during exceptional circumstances. It works well enough if the two

[16] In a contracting dynamics, nearby points are driven closer by the dynamics.

[17] The $\tilde{1}$ refers to an L_1 norm on the (instantaneous values of the) tilde variables.

systems quickly recover ‘nearness’, or if the models cease to be relevant at all after the exception.

In our case, the ‘exceptional’ regime, requiring pulses, is precisely the *raison d’etre* of the whole protection system, and it is in this regime (rather then the normal, stable regime when there is no earthquake) in which the behaviours of the II and SQ systems are the most unruly. And although retrenchment, as described in [14, 13, 30], addresses the onset of unruly behaviour quite well, it does not really engage with particular properties of extended periods of unruly behaviour, as we would ideally like in our application.

A further complication of the scenario where the SQ system pulses and the II system does not, is that different events in the two systems are involved in these behaviours (*PulseNo* and *PulseMaybe* in II and *PulseYesY_S* and *PulseYesE_S* in SQ). Retrenchment and refinement, as usually defined, assume a static (and partial if needed) bijection between operations/events in the two models being compared. This does not cope with the scenario just mentioned, in which overlapping pairs of (names of) events may need to be related at different times.

Thus our reticence in writing down an explicit level 3 system (with its obligation to make clear its relationship to the level 2 system) is further explained by the absence of a suitable species of formal relationship that could be used for the purpose. Without getting embroiled in too many further details, the present case study provides a fertile stimulus for developing a richer formulation of retrenchment and refinement capable of coping with the wealth of phenomena it exhibits.

11 Experiments and Simulations

In this section, we compare the expectations raised by the preceding analytical work, with the outputs of well established conventional earthquake protection design approaches, based on numerical simulation.

Our simulations were performed over a time interval from 0 to $\mathcal{T}_{\mathrm{MAX}}$, using a control strategy that, as suggested by the analytical work, is defined by a pulse interval T_P, allowable relative displacement X_B, and an additional ground acceleration variability term e_B. At the start of a pulse interval, the simulation chooses whether to apply a pulse based on equation (14), predicting a value of x at the end of the interval from the expression $h_x x + h_y y + h_e (e \pm e_B)$, where:

$$h_x = 1 - \frac{\omega^2 T_P^2}{2} \qquad h_y = T_P \Big(1 - \zeta\, \omega T_P\Big) \qquad h_e = -\frac{T_P^2}{2} \tag{19}$$

using actual values of —or available estimates for— x, y, and e at time t.

Fig. 15 shows the essence of a Python program that performs simulations using the numerical and scientific libraries `NumPy` and `SciPy`, as well as the `Matplotlib` library for plotting; the complete code is available online [8].

```
1  def simulate($\mathcal{T}_{\mathrm{MAX}}$, $T_P$, $X_B$, $e_B$):

2      def x_future(x, y, t):
3          f = lambda s: $h_x$ * x + $h_y$ * y + $h_e$ * (e(t) + s * $e_B$)
4          a, b = f(1), f(−1)
5          return (a, 1) if abs(a) > abs(b) else (b, −1)

6      def y_new(x_desired, x, t, s):
7          return (x_desired − $h_x$ * x − $h_e$ * (e(t) + s * $e_B$)) / $h_y$

8      x, y, t = $x_0$, $y_0$, 0                    # initial condition
9      for i in range(int($\mathcal{T}_{\mathrm{MAX}}$/$T_P$)):
10         xe, sign = x_future(x, y, t)
11         if abs(xe) > $X_B$:
12             y = y_new(signum(xe) * $X_B$, x, t, sign)
13         x, y, t = advance(x, y, t, (i + 1) * $T_P$)
```

Fig. 15 Python program for numerical simulation [8].

Function **simulate**($\mathcal{T}_{\mathrm{MAX}}, T_P, X_B, e_B$) contains two nested functions, one to predict future values of x, and another to adjust current values of y, if needed, when a pulse is called for. In particular, function **x_future**(x,y,t) estimates x at a time $t+T_P$ in the future, returning the estimate and the sign used for the e_B term that maximizes the absolute value of the estimate — the worse case. The value returned by function $e(t)$ is the ground acceleration at time t. Function **y_new**(x,y,t) likewise uses equation (14), but in this case does so to find a new value of y that would, one hopes, cause $|x(t+T_P)| \leq X_B$ to be satisfied; the sign for e_B must be supplied (in this case, by the result from **x_future**).

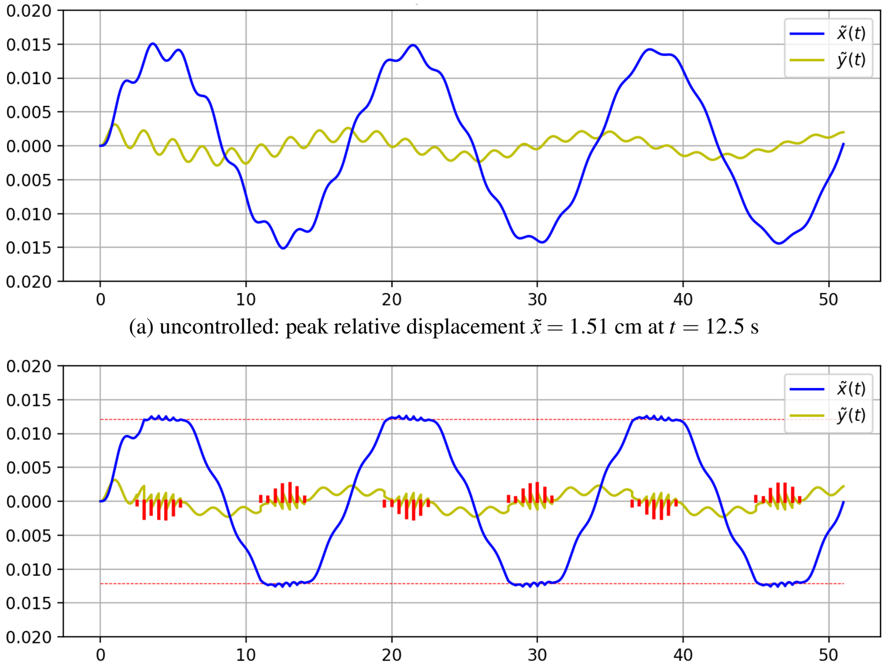
As with the HEB model, the simulation (defined by lines 8–13) is broken up into a succession of subintervals, each with duration T_P. Between subintervals, a pulse may be applied that changes the value of y instantaneously. During a subinterval, time marches from $i\,T_P$ to $(i+1)\,T_P$.[18] Function **advance**(x,y,a,b), not shown, lets the system evolve from time a to time b, starting from the initial values $x(a)$ and $y(a)$; it returns their values at time b: $x(b)$, $y(b)$, and b. As a side effect, it builds up collections of data for plotting time histories of x and y.

With respect to numerical integration, *advance* solves the system of first order differential equations:

$$\mathcal{D}x = y \tag{20}$$

$$\mathcal{D}y = -2\zeta\,\omega_n\,y - \omega_n^2\,x - e(t) \tag{21}$$

[18] The form *for* i *in range*(n) is idiomatic Python for bounded iteration from 0 to $n-1$ (inclusive).



(a) uncontrolled: peak relative displacement $\tilde{x} = 1.51$ cm at $t = 12.5$ s

(b) controlled at $X_B = 80\%$ of peak relative displacement

Fig. 16 Response to harmonic ground motion ($T_n = 2$ s, $\zeta = 1\%$, $Z = 1$, $\Omega = 0.37$ rad/s).

introduced in equation (3) and subsequently redefined in LHS (4) in terms of the variables: ζ, the viscous damping factor (dimensionless fraction of critical damping); and ω_n, the undamped circular natural frequency (in units of radians per second). It does so using `odeint`, a `SciPy` function based on the Fortran `LSODA` routine from the `ODEPACK` library, which uses an Adams predictor-corrector method (when non-stiff problems like ours are encountered). The routine determines step size automatically to ensure that error bounds are satisfied.

Harmonic ground motion. To illustrate the approach, we begin with a simple example after Prucz et al. [29] of an SDOF system, like that of Fig. 3, with a natural frequency $\omega_n = \pi$ rad/s and viscous damping factor $\zeta = 1\%$. It is subjected to harmonic ground motion $z = Z \sin \Omega t$, which has the effect of adding a reversed inertia force $-m\mathcal{D}^2 z$ to the system, with the ground acceleration given by:

$$e(t) \equiv \mathcal{D}^2 z = -\Omega^2 Z \sin \Omega t \tag{22}$$

where amplitude $Z = 1$ and frequency $\Omega = 0.37$ rad/s are given. Thus, the case is one in which the ground motion frequency is lower than the system natural frequency (i.e., $\Omega < \omega_n$). The system begins at rest, so $x_0 = y_0 = 0$.

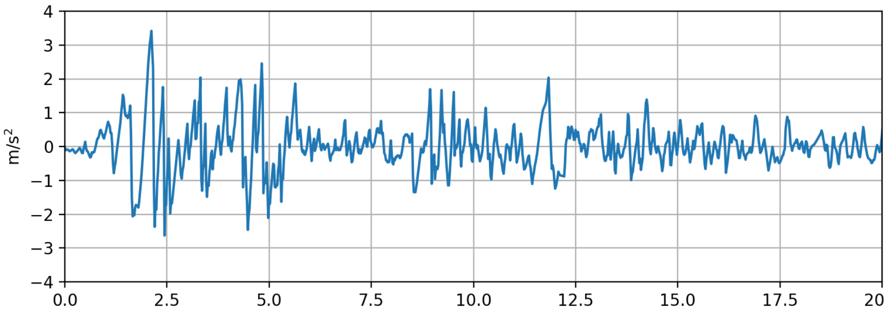


Fig. 17 North-south component of the ground motion recorded at a site in El Centro, California, during the Imperial Valley earthquake of May 18, 1940 (showing first 20 s of the event).

Time histories of the uncontrolled response are shown in Fig. 16.(a), where the dimensionally comparable quantities $\tilde{x} \equiv x$ and $\tilde{y} \equiv y/\omega_n$ are plotted. The peak responses are:

$$\tilde{x}(12.5 \text{ s}) = -0.0151, \quad \tilde{y}(0.985 \text{ s}) = 0.00315$$

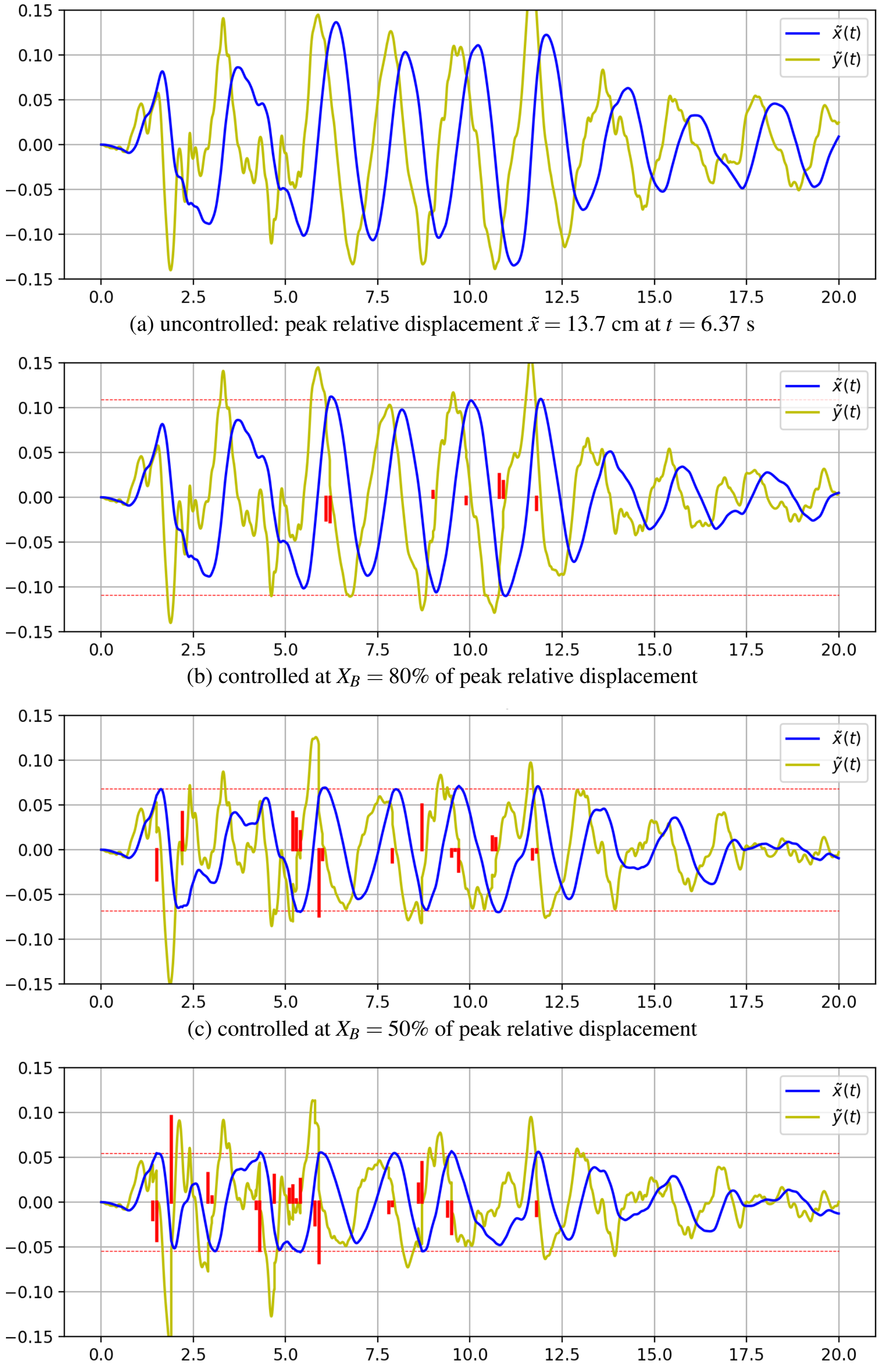
The predominant response of $\tilde{x}(t)$ is a harmonic having the same frequency as that of the ground acceleration; its period is $2\pi/\Omega$, or in this case about 17 s. As expected, when $\Omega \ll \omega_n$ there is little relative motion between the mass and the ground, and the motions are in phase: they reach their peaks at the same time. Superimposed 'wiggles' are (dying) transients induced at the natural frequency of the system, whose undamped natural period $T_n = 2$ s.

Pulse control can now be employed to limit the response to 80% of the peak relative displacement, which is done by setting $X_B = 0.0121$. Continuing to be consistent with Prucz et al., we set the pulse interval to be on the order of one fourth the natural period, so $T_P = T_n/4 = 0.5$ s. Specific to our approach, the additional ground acceleration variability term e_B is set to zero for the moment. Time histories of the controlled response are shown in Fig. 16.(b), where the pulse trains (in red) have a 'shape' that acts to counterbalance relative displacements, where needed, that would have occurred, so as to keep them roughly within desired limits.

Though the pulse interval here is about five times larger than we would anticipate using in practice, the example motivates the definition of a metric, the *exceedance level*, that can be used to assess the algorithm's effectiveness as a bounded state control strategy. To quantify the exceedance level, we consider what happens at the endpoint of a T_P interval where, if $|x(t)| > X_B$, we add $|x(t)| - X_B$ to a running sum $\mathcal{S}$, and define:

$$\mathcal{E} = 10^3\, \mathcal{S} \,/\, n X_B \tag{23}$$

where n is the number of pulse intervals included in sum $\mathcal{S}$. For the Prucz example, that gives an exceedance level $\mathcal{E} = 9.07$. For pulses at 20 times per natural period



(a) uncontrolled: peak relative displacement $\tilde{x} = 13.7$ cm at $t = 6.37$ s

(b) controlled at $X_B = 80\%$ of peak relative displacement

(c) controlled at $X_B = 50\%$ of peak relative displacement

(d) controlled at $X_B = 40\%$ of peak relative displacement

Fig. 18 Response to El Centro ground motion ($T_n = 2$ s, $\zeta = 5\%$, no time delay).

instead (i.e., for $T_P = 0.1$ s), we have $\mathcal{E} = 0.0250$, and when in addition e_B is raised to 0.001, the exceedance level $\mathcal{E}$ drops to zero, meaning there are no exceedances.

El Centro ground motion. As noted by Prucz et al., the aim of pulse control is to disrupt, at resonance, the 'gradual rhythmic build-up' of the system response. A more realistic and challenging scenario then is to subject the system to complex ground accelerations that include resonant frequencies, particularly ones near the fundamental natural frequency of a building, which typically produce the largest relative displacements and damage.

Used in the design of earthquake resistant structures, the ground accelerations recorded in El Centro, California, during the earthquake of May 18, 1940, have a peak value of 3.13 m/s^2 (0.319 g), the first 20 s of which are shown in Fig. 17. We now apply them to the system. As before, the natural frequency $\omega_n = \pi$ rad/s, so the undamped natural period $T_n = 2$ s, a value that might correspond to the fundamental natural period of a 20-story building. We use a viscous damping factor $\zeta = 5\%$, which is representative of a modern office building and is a value often used in design. For control, the pulse interval $T_P = 0.1$ s.

Time histories of the uncontrolled response are shown in Fig. 18 (a), where we again plot $\tilde{x}$ and $\tilde{y}$. The peak responses are

$$\tilde{x}(6.37\ \text{s}) = 0.137, \quad \tilde{y}(11.7\ \text{s}) = 0.199$$

which occur during a time period from about 6 to 13 seconds into the event, as the system begins oscillating near its undamped natural frequency $\omega_n = \pi$ rad/s (with a period $T_n = 2$ s).

To limit the peak displacement, we apply pulse control at 80% of that value by setting $X_B = 0.109$ (or 10.9 cm) and keep the pulse interval as before, $T_P = 0.1$ s. Time histories of the controlled response are shown in Fig. 18.(b), where the pulses, shown in red, effectively counterbalance relative displacements to keep them approximately within desired bounds. Limiting displacements even further, to 50% and 40% of the peak value, is likewise shown to be effective, as demonstrated by the time histories in Figs. 18.(c)-(d), respectively. To achieve the additional level of control requires that successively more energy be put into the system, with more and sometimes larger pulses, and earlier into the event.

Looking at exceedance for the three levels of controlled response (80%, 50%, and 40%), we have $\mathcal{E} = 0.223$, 0.664, and 0.561, respectively, which are reduced to zero when the additional ground acceleration term, e_B, is increased to at least 0.512, 0.358, and 0.469, respectively. Additional analysis, that might lead to finding good e_B settings *a priori* for anticipated ground motions, is left for future work.

12 Conclusions

In this paper, we started by reviewing how the initial ideas of earthquake protection eventually crystallised into a number of distinct approaches, and we focused on the active control approach. We also reviewed Hybrid Event-B as a suitable vehicle for attempting a formal development of an SDOF active protection model. We then pursued the development through various levels of detail, culminating in the distributed sampled and quantized model of Section 9. Along the way, particularly in Sections 8 and 10, we discussed the obstacles to accomplishing this with full formality.

In Section 11, we subjected our analytically derived model to simulation using well established numerical tools typically used in earthquake protection engineering. We spot-tested our model both on a simple harmonic excitation, and on the El Centro ground motion data. It was encouraging to see that our model behaved well, despite the relatively small input from the empirical sphere during its derivation. Enhancing the latter, can only be expected to improve matters regarding fidelity with conventional approaches.

The present study forms a launchpad for much possible future work. Firstly, there is the fact that our models' behaviour was timed with precision — in reality we will always have stochastic variations in the times of all events. Similar considerations apply to a better characterisation of the additional ground acceleration variability term e_B. Taking these issues into account would bring us closer to the level of detail of [16, 20, 21].

Secondly, there is the consideration of the replication of components needed for adequate fault tolerance. Here, at least, we can see that use of standard approaches would address the issue, and would again bring us closer to [16, 20, 21].

Thirdly, we note that the SDOF modelling can readily be enriched to capture the dynamics of a genuine building more accurately. The essentially scalar description we dealt with here could be enriched to encompass a greater number of linear and angular degrees of freedom. This again is relatively standard, at least in the linear dynamics case.

Fourthly, there is the investigation of richer formulations of retrenchment and refinement capable of coping with the wealth of phenomena discussed in Section 10. A generally applicable approach here would yield many dividends for a wide class of problems of a similar nature.

Fifthly, it is regrettable that there currently is no mechanised support for Hybrid Event-B. Nevertheless, progress with the issue just discussed would be a prerequisite for a meaningfully comprehensive coverage of the development route as a whole by mechanical means, even if individual parts could be treated by conventional mechanisation of linear and discrete reasoning. Taking all the above together, there is plenty to pursue in future work.

One final comment. In a recent UK terrestrial TV broadcast [25], various aspects of the construction of Beijing's Forbidden City were described. Not least among these was the capacity of the Forbidden City's buildings to withstand earthquakes,[19] particularly considering that Beijing lies in a highly seismic region. Fundamental to this is the use of bulky columns, which are essentially free standing, to support the weight of the building's heavy roof, and the use of complex dougong brackets [37, 38] to couple the columns to the roof. The free standing construction allows the ground under the building to slip during powerful tremors without breaking

[19] Being wooden, the Forbidden City's buildings were less good at withstanding fire, and several structures have had to be rebuilt a number of times over the centuries because they had burnt down.

the columns, and the relatively flexible dougong brackets permit relative movement between the columns and other members without risking structural failure. These building techniques were already ancient by the time the Forbidden City was constructed early in the 1400's. The cited broadcast showed a scaled structure on a shaking table withstanding a simulated magnitude 10 quake. So, more recent efforts notwithstanding, the Chinese had the problem of earthquake protection for buildings licked more than two thousand years ago!

References

1. Journal of Earthquake Engineering and Engineering Vibration.
2. World Conferences on Earthquake Engineering.
3. Abrial, J.R.: Modeling in Event-B: System and Software Engineering. Cambridge University Press (2010)
4. Ahmed, N.: Dynamic Systems and Control With Applications. World Scientific (2006)
5. Banach, R.: Formal Refinement and Partitioning of a Fuel Pump System for Small Aircraft in Hybrid Event-B. In: Bonsangue, Deng (eds.) Proc. IEEE TASE-16, pp. 65–72. IEEE (2016)
6. Banach, R.: Hemodialysis Machine in Hybrid Event-B. In: Butler, Schewe, Mashkoor, Biro (eds.) Proc. ABZ-16, vol. 9675, pp. 376–393. Springer, LNCS (2016)
7. Banach, R.: The Landing Gear System in Multi-Machine Hybrid Event-B. Int. J. Soft. Tools for Tech. Trans. **19**, 205–228 (2017)
8. Banach, R., Baugh, J.: Active earthquake control case study in Hybrid Event-B web site. http://www.cs.man.ac.uk/~banach/some.pubs/EarthquakeProtection/
9. Banach, R., Butler, M.: A Hybrid Event-B Study of Lane Centering. In: Aiguier, Boulanger, Krob, Marchal (ed.) Proc. CSDM-13, pp. 97–111. Springer (2013)
10. Banach, R., Butler, M.: Cruise Control in Hybrid Event-B. In: Z. Liu Woodcock (ed.) Proc. ICTAC-13, *LNCS*, vol. 8049, pp. 76–93. Springer (2013)
11. Banach, R., Butler, M., Qin, S., Verma, N., Zhu, H.: Core Hybrid Event-B I: Single Hybrid Event-B Machines. Sci. Comp. Prog. **105**, 92–123 (2015)
12. Banach, R., Butler, M., Qin, S., Zhu, H.: Core Hybrid Event-B II: Multiple Cooperating Hybrid Event-B Machines. Sci. Comp. Prog. **139**, 1–35 (2017)
13. Banach, R., Jeske, C.: Retrenchment and Refinement Interworking: the Tower Theorems. Math. Struc. Comp. Sci. **25**, 135–202
14. Banach, R., Poppleton, M., Jeske, C., Stepney, S.: Engineering and Theoretical Underpinnings of Retrenchment. Sci. Comp. Prog. **67**, 301–329 (2007)
15. Banach, R., Van Schaik, P., Verhulst, E.: Simulation and Formal Modelling of Yaw Control in a Drive-by-Wire Application. In: Proc. FedCSIS IWCPS-15, pp. 731–742 (2015)
16. Baugh, J., Elseaidy, W.: Real-Time Software Development with Formal Methods. J. Comp. in C. Eng. **9**, 73–86 (1995)
17. Buckle, I.: Passive Control of Structures for Seismic Loads. In: Proc. 12th World Conference on Earthquake Engineering (2000). Paper No. 2825
18. Chicone, C.: Ordinary Differential Equations with Applications, 2nd edn. Springer (2006)
19. Chopra, A.: Dynamics of Structures: Theory and Applications to Earthquake Engineering, 4th. edn. Pearson (2015)
20. Elseaidy, W., Baugh, J., Cleaveland, R.: Verification of an Active Control System Using Temporal Process Algebra. Eng. with Comp. **12**, 46–61 (1996)
21. Elseaidy, W., Cleaveland, R., Baugh, J.: Modeling and Verifying Active Structural Control Systems. Sci. Comp. Prog. **29**, 99–122 (1997)
22. Gattulli, V., Lepidi, M., Potenza, F.: Seismic Protection of Frame Structures via Semi-Active Control: Modelling and Implementation Issues. Eq. Eng. Eng. Vib. **8**, 627–645 (2009)

23. Gradshteyn, I., Ryzhik, I.: Table of Integrals Series and Products, 7th edn. Academic Press (2007)
24. Kuramoto, H., Teshigawara, M., Okuzono, T., Koshika, N., Takayama, M., Hori, T.: Predicting the earthquake response of buildings using equivalent single degree of freedom system. In: Proceedings of 12th World Conference on Earthquake Engineering. Auckland, New Zealand (2000). Paper No. 1039
25. More4 TV: Secrets of China's Forbidden City. (24 July 2017). UK Terrestrial TV Channel: More4.
26. Olver, F., Lozier, D., Boisvert, R., Clark, C.: NIST Handbook of Mathematical Functions. Cambridge University Press (2010)
27. Platzer, A.: Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics. Springer (2010)
28. Popescu, I., Sireteanu, T., Mitu, A.: A Comparative Study of Active and Semi-Active Control of Building Seismic Response. In: Proc. DGDS-09, pp. 172–177. Geometry Balkan Press (2010)
29. Prucz, Z., Soong, T., Reinhorn, A.: An analysis of pulse control for simple mechanical systems. J. Dyn. Sys. Meas. Cont. **107**, 123–131 (1985)
30. Retrenchment Homepage: http://www.cs.man.ac.uk/~banach/retrenchment
31. Rose, B., Baugh, J.: Parametric Study of a Pulse Control Algorithm with Time Delays. Tech. Rep. Technical Report CE-302-93, North Carolina State University Department of Civil Engineering (1993)
32. Sontag, E.: Mathematical Control Theory. Springer (1998)
33. Soong, T.: Active Structural Control: Theory and Practice. Longman (1990)
34. Soong, T., Chu, S., Reinhorn, A.: Active, Hybrid and Semi-Active Control: A Design and Implementation Handbook. Wiley (2005)
35. Tabuada, P.: Verification and Control of Hybrid Systems: A Symbolic Approach. Springer (2009)
36. Walter, W.: Ordinary Differential Equations. Springer (1998)
37. Wikipedia: Chinese architecture.
38. Wikipedia: Dougong.
39. Wikipedia: Duhamel's Integral.
40. Wikipedia: Earthquake Engineering.