Graded Refinement, Retrenchment and Simulation

RICHARD BANACH, University of Manchester, UK

Refinement of formal system models towards implementation has been a mainstay of system development since the inception of formal and Correct by Construction approaches to system development. However, pure refinement approaches do not always deal fluently with all desirable system requirements. This prompted the development of alternatives and generalisations, such as retrenchment. The crucial concept of simulation is key to judging the quality of the conformance between abstract and more concrete system models. Reformulations of these theoretical approaches are reprised, and are embedded in a graded framework. The added flexibility this offers is intended to deal more effectively with the needs of applications in which the relationship between different levels of abstraction is not straightforward, and in which behaviour can oscillate between conforming quite closely to an idealised abstraction, and deviating quite far from it. The framework developed is confronted with an intentionally demanding case study: a model active control system for the protection of buildings during earthquakes. This offers many challenges: it is hybrid/cyber-physical; it has to respond to rather unpredictable inputs; it has to straddle the gap between continuous behaviour and discretized/quantized/numerical implementation.

CCS Concepts: • **Software and its engineering → Formal methods**; **Software verification and validation**; **Formal software verification**; • **Computer systems organization** → *Embedded and cyber-physical systems*;

Additional Key Words and Phrases: Refinement, Retrenchment, Simulation

ACM Reference Format:
Richard Banach. 2099. Graded Refinement, Retrenchment and Simulation. *ACM Trans. Softw. Eng. Methodol.* 999, 4, Article 9876 (March 2099), 67 pages. https://doi.org/0000001.0000001

1 INTRODUCTION

Refinement of formal system models towards implementation has been a mainstay of system development since the inception of formal and Correct by Construction approaches to system development. However, pure refinement approaches have not always been able to deal fluently with all the system requirements that may be desired in some particular application. This observation prompted the development of alternatives and generalisations, such as retrenchment.

Retrenchment [14, 15, 21, 66], was originally introduced in [19] (in the context of the B-Method [2, 3]) to capture system development steps that do not fit comfortably within standard refinement pathways. Subsequently it developed a substantial literature showing the convenience and utility of the technique. Among the more notable highlights of this, was the work done on the Mondex Purse [73], an electronic wallet application designed to enable exchange of value electronically — nowadays a familiar and widespread capability but quite novel at the time of its introduction.

In the Mondex Purse, retrenchment was applied in many ways to capture how a more realistic model of the implementation could be related to the idealised formal models in [73]. Thus, in [20],

Author's address: Richard Banach, University of Manchester, Department of Computer Science, Oxford Road, Manchester, M13 9PL, UK, richard.banach@manchester.ac.uk.

1049-331X/2099/3-ART9876 $15.00
https://doi.org/0000001.0000001

the sequence numbers of Mondex transactions had a finite (though very large) upper limit, whereas in [73] they were unbounded natural numbers. In [16] the size of the Mondex exception log was finite (and decidedly small), whereas in [73] it was unbounded — this required an approach different from the previous case. In [17] the non-injectivity of the Mondex CLEAR codes was treated. The CLEAR codes embody permission from a central archive to a Mondex purse to clear its exception log without loss to any part of the system. In [73], the CLEAR code is injectively matched to the log contents whereas in reality, it is a hash, opening the possibility of a hash collision and thence of loss to some user — but crucially, not the creation of new value, the key security property. Finally, in [18], retrenchment was used to resolve an incongruity in the *BalanceEnquiry* operation of the Mondex protocol, arising from different atomicity/granularity properties of different abstraction levels of the models involved. (In fact the incongruity was acute enough that the *BalanceEnquiry* was not included in the published models.) Mondex itself was the focus of a major *Verification Grand Challenge* [52, 53, 79, 80]. A more recent overview of embeddings of the retrenchment idea into various formal development frameworks appears in [10].

Retrenchment, as originally presented, works very well when the deviation between idealised and realistic models concerns a system failure, departing from a hitherto established refinement situation from which recovery is not contemplated — or when its additional flexibility can be used to hide temporary departures from refinement that clash with the detailed requirements of a particular refinement framework. The Mondex case studies were primarily of this type. Retrenchment works less well in cases where the departures from refinement are significant — but where they are nevertheless recovered from in various ways to reestablish a refining state of affairs between idealised and realistic models. In such situations, while the departures from refinement can be well described by retrenchment, the recovery situation is less well covered.

Typical examples of systems of the latter kind are found among hybrid and cyber-physical systems [28, 38]. To see the relevance of this, imagine a cyber-physical system in which an idealised model captures continuous behaviour using conventional ODE systems, but a more realistic and concrete model works using numerical algorithms to give algorithmic implementations of solutions to the ODE systems. In this situation, the inevitable deviations between the idealised ODEs and their numerical (and thus inevitably approximate) implementations, mean that when changes of system behaviour are prompted by continuous variables crossing threshold values, such threshold crossings can take place at different points in the two models —leading to widely differing behaviours from a refinement viewpoint— even if the overall behaviour in the two systems converges later, for example as a consequence of system stability considerations.

This paper aims to reappraise the original retrenchment ideas in order to furnish a formalism that can cope with such situations a lot more fluently. In addition to that, there is another issue that arises that is novel when compared with conventional stepwise refinement frameworks. In the conventional case, in a stepwise refinement development step, the abstract system model is typically regarded as being 'prior', in the sense that the requirements, properties and invariants it captures should be maintained in a suitable manner further down the development hierarchy — and in the concrete model of the development step in particular: some mild technical conditions are usually enough to guarantee that. The conflation of 'prior' with 'abstract' and the consequent conflation of 'non-prior' with 'concrete' correlates with the direction of the implication in typical refinement or retrenchment correctness proof obligations: for each non-prior behaviour there must be a prior one that it concretises.

In the hybrid and cyber-physical systems case, the prior model would naturally be the one containing physical law and the intended continuous design — since that is what corresponds to the unavoidable natural world. But we have just seen that concrete numerical models cannot maintain continuous physical law with precision in general. The alternative then, would be to view

the concrete numerical model as prior, because it captures implementation issues connected with discretisation that are unavoidable — it being necessary that the abstract model not make demands that cannot be fulfilled by the concrete model. The latter requirement suggests conflating 'prior' with 'concrete'. This would make the continuous, physical model a refinement or retrenchment of the numerical one.

Neither approach seems ideal —there is a palpable tension between the two views— and in this setting, we can find that notions based on simulation, offering a more even handed balance between abstract and concrete, are better suited than the more traditional and more directed notions of refinement and retrenchment, with their insistence that one model conforms completely to the exigencies of the other. In a sense, this refocuses the relationship between the two models in a more angelic direction, in contrast to the more directed demonic notions. The work in this paper elaborates this perspective in a technically more detailed direction.

The remainder of the paper is as follows. Section 2 presents our system concepts. The approach in this portion of the paper is primarily semantic, i.e. it is based on transition system concepts. Thus, specific languages and syntactic frameworks are not considered yet. Our formulation of refinement for these systems follows in Section 3. Section 4 addresses retrenchment, significantly reformulated compared with earlier work. Section 5 says what needs to be done to accommodate continuously varying state change in these frameworks, given that hybrid and cyber-physical systems are so important, both for the main case study of this paper, and in general. Section 6 covers the many notions of simulation that the preceding ideas admit.

Section 7 presents some straightforward results on departure from and return to conformance with desired invariants, in cases where absolute adherence to them cannot be guaranteed. These are based on quite strong assumptions. Section 8 tackles the same issue, but this time basing the approach on metric assumptions about the systems in question, and on suitable contracting properties. Section 9 discusses the implications of ensuring the preceding results extend also to continuous behaviours, building on the approach of Section 5.

Then Section 10 presents the graded integration of the preceding ideas, contributing an overall focus to the paper: the graded development system.

Section 11 introduces the essentials of Hybrid Event-B [12, 13], the formalism within which the main motivating case study in the paper is expressed. Section 12 then introduces the case study itself. This concerns an active control system for earthquake protection, first discussed in [11]. As well as involving discrete and continuous behaviours in an essential way, the 'exceptional' parts of the system's behaviour are highly important from the system development viewpoint, and thus merit being treated as 'first class citizens' in the development process, via the theory developed earlier. Although the investigation of the earthquake protection system is expressed using the syntax of Hybrid Event-B, the semantics of refinement and retrenchment that is utilised, is the semantics developed earlier in this paper. We make suitable comments as needed. Sections 11 and 12 can be read directly after this Introduction, as they do not depend on the intervening material.

While Section 12 contents itself with describing the models of the earthquake protection system development, Section 13 considers how they are related to one another from the more formal vantage points developed earlier. This is a large section, divided into many subsections, which explores the relationships between the models from a wide variety of possible perspectives. With this in hand, Section 14 considers how the whole of the preceding can be viewed as an example of a graded development system. Section 15 broadens the discussion and considers some related work. Section 16 concludes. It encompasses discussion of how the ideas developed here can be applied in a wider context, and what the impact of such ideas would be for tool support.

The level of detail needed to make the concepts developed in this paper convincing, make the paper rather long. Readers may wish to skip over the proofs on a first reading.

Assumption 1.1. *We work in a set theoretic and relational framework, in which relations are manipulated using logical operations on the predicates that define their bodies. We are not pedantic about distinguishing a set or relation from the predicate that defines it. To avoid a proliferation of pathological cases, we assume henceforth, that any set or relation mentioned in the hypotheses of a construction or theorem is nonempty, so that, for example, a mentioned putative choice of some element from it can actually be made.*

2 BASIC SYSTEMS

In this section we give our basic definitions and notations for the theoretical framework we will subsequently build. We will deal with transition systems. A typical system will be $SysX$, where the label X distinguishes the system from other transition systems in the discourse. $SysX$ has a set of operation names Ops_X, with typical element Op_X. An operation Op_X works on the state space U_X with typical element u_X. Op_X will also have an input space I_{Op_X} with typical element i_X, and an output space O_{Op_X} with typical element o_X.[1] An individual step, or transition of Op_X is typically written as $u_X \text{-}(i_X, Op_X, o_X)\!\!\rightarrow u'_X$, where u'_X is the after-state. Their totality constitutes the step, or transition relation $stp_{Op_X}(u_X, i_X, u'_X, o_X)$ of Op_X. When we aggregate the transition relations of all the operations Op_X of $SysX$, we obtain the complete transition relation for the $SysX$ system, $stp_X = \bigcup_{Op_X \in \mathrm{Ops}_X} stp_{Op_X}$, where the union is necessarily disjoint since the relevant Op_X name is part of every execution step. An initial state of $SysX$ is assumed to be defined by a predicate $Init_X(u_X)$ — inputs and/or outputs are only needed once a transition occurs. Summarising, and suppressing the label X, a system Sys can be seen as a tuple with signature[2] $(\mathrm{Ops}, \mathsf{U}, Init, \uplus_{Op \in \mathrm{Ops}} \mathsf{I}_{Op}, \uplus_{Op \in \mathrm{Ops}} \mathsf{O}_{Op}, stp : \uplus_{Op \in \mathrm{Ops}} (\{Op\} \times \mathsf{U} \times \mathsf{I}_{Op} \times \mathsf{U} \times \mathsf{O}_{Op}))$.

Accompanying every system $SysX$, we will assume that there can be a state invariant, Inv_X. This mechanism can be used to provide a safe, useful subsystem of $SysX$. For this to work, it is sufficient that the initial state(s) satisfy Inv_X:

$$Init_X(u_X) \Rightarrow Inv_X(u_X) \tag{1}$$

and that the stp_X relation preserves satisfaction of Inv_X:

$$Inv_X(u_X) \wedge stp_{Op_X}(u_X, i_X, u'_X, o_X) \Rightarrow Inv_X(u'_X) \tag{2}$$

The subset of transitions $u_X \text{-}(i_X, Op_X, o_X)\!\!\rightarrow u'_X$ whose states u_X and u'_X satisfy $Inv_X(u_X)$ and $Inv_X(u'_X)$ define the *invariant* subsystem $SysInvX$, consisting of invariant transitions $Invstp_X$, and its states $Inv\mathsf{U}_X$ (defined by the predicate Inv_X), and its I/O values $Inv\mathsf{I}_{Op_X}$ and $Inv\mathsf{O}_{Op_X}$ for the various Op_X. Invariants of this kind are very convenient for expressing safety properties of the system dynamics.

A useful variation of the notion of invariant is the idea of a contingent invariant. For a system $SysX$, a contingent invariant $CInv_X$ with respect to a safe set of inputs $CInv\mathsf{I}_{Op_X} \subseteq \mathsf{I}_{Op_X}$ satisfies:

$$Init_X(u_X) \Rightarrow CInv_X(u_X) \tag{3}$$

$$CInv_X(u_X) \wedge i_X \in CInv\mathsf{I}_{Op_X} \wedge stp_{Op_X}(u_X, i_X, u'_X, o_X) \Rightarrow CInv_X(u'_X) \tag{4}$$

Thus, while an invariant holds regardless of system behaviour (i.e. our formalism does not permit the idea of *ceasing to* conform to an invariant), a contingent invariant may fail if the system is driven beyond 'reasonable bounds' by receiving extraordinary inputs.

[1] We allow the input and output spaces to depend on the operation name, but we normally suppress this dependence in the notation. Moreover, if inputs and/or outputs are not needed, we can simply elide mentioning them in the text and formulas below.

[2] $\uplus$ denotes disjoint union.

The subset of transitions u_X -(i_X, Op_X, o_X)↣ u'_X whose inputs satisfy $i_X \in CInvI_{Op_X}$ and whose states u_X and u'_X satisfy $CInv_X(u_X)$ and $CInv_X(u'_X)$ define the *contingent invariant* subsystem (with respect to $CInvI_{Op_X}$) *SysCInvX*, consisting of invariant transitions $CInvstp_X$, and its states $CInvU_X$ (defined by the predicate $CInv_X$), and its I/O values $CInvI_{Op_X}$ and $CInvO_{Op_X}$ for the various Op_X. Contingent invariants are useful for expressing safety properties of the system dynamics that are expected to hold except in exceptional circumstances.

None of the systems discussed so far is compelled to be reachable in its entirety. Thus, if we restrict the transition relation for *SysX*, stp_X, to just those transitions u_X -(i_X, Op_X, o_X)↣ u'_X whose states u_X (and u'_X) are accessible via a finite number of steps from an initial state, we get *ReachSysX*, the subsystem consisting of *reachable* transitions $Reachstp_X$, and its states $ReachU_X$, together with its I/O values $ReachI_{Op_X}$ and $ReachO_{Op_X}$ for the various Op_X. The same consideration may be applied to *SysInvX* and to *SysCInvX*, yielding their reachable subsystems *ReachSysInvX* and *ReachSysCInvX*.

Given our machinery so far, we can thus contemplate six systems: *SysX*, as originally introduced, and its reachable subsystem *ReachSysX*; *SysInvX*, an invariant subsystem with respect to Inv_X, and its reachable subsystem *ReachSysInvX*; *SysCInvX*, a contingent invariant subsystem whose state predicate $CInv_X$ may exceptionally be violated if inputs do not satisfy $i_X \in CInvI_{Op_X}$, and its reachable subsystem *ReachSysCInvX*.

Evidently, all are subsystems of *SysX* (if only trivially for *SysX* itself). By a simple induction based on (1) and (2), *ReachSysX* = *ReachSysInvX*, and by a similar induction, *ReachSysX* is a subsystem of *SysInvX* (and is trivially a subsystem of itself). Given some $CInv_X$, neither of *SysCInvX* or *ReachSysX* need be a subsystem of the other in general, but *ReachSysCInvX* is always a subsystem of *ReachSysX*, and if $CInv_X \Rightarrow Inv_X$ then *SysCInvX* is a subsystem of *SysInvX*.

A system other than *SysX* will have a similar structure, being distinguished by a different label, say *Y*, thus: *SysY*, with all the other elements labelled analogously. In Section 10, the labels will form a partial order, but for now, we introduce the relationships between a pair of systems that we regard as being in the scope of the current discourse. We describe these in terms of the original *SysX* notion, but the same ideas can be applied to the various subsystems just introduced, as needed.

3 REFINEMENT

Refinement is a way of developing a desired system in stages, with the stages being constrained by some formal stipulations. These allow some claimed properties of the process to be verified, and experience has shown that this is beneficial to the accuracy of the development methodology. The original concept is old (see e.g., [33, 34, 71, 81]), and many variations on the same basic idea exist in the literature. Most of this section is an adaptation of known ideas to the systems framework of Section 2.

Suppose *X* is the label of system *SysX* and *Y* is the label of system *SysY*, and suppose that these are successive stages in a refinement development. To say that *SysX* is *refined by* (we also say refined to) *SysY* —written *SysX* ⩾ *SysY* in this paper— signifies the following collection of facts.

Firstly, there exists a gluing relation between the state spaces $G : U_X \leftrightarrow U_Y$.[3] No technical restriction is placed on its content *a priori* in the approach of this paper. The gluing relation expresses what the design of the refinement strategy considers important enough to be worthy of formal control between the abstract model *SysX*'s state space and the concrete model *SysY*'s state space, and it is important to remember that this is a *human* decision. None of the formal machinery described here (or indeed, in any formal approach) can absolve humans from the responsibility of deciding *what* the developed system is supposed to do and how that is to be achieved. It can only

[3]The gluing relation is also referred to as a refinement relation, or a retrieve relation in other work on refinement.

help them to do the requisite development better. And the same observation applies to all the other data items relevant to the formal discourse in this paper.

Secondly, given G, an initialisation proof obligation (PO) holds:

$$Init_Y(u_Y) \Rightarrow (\exists\, u_X \bullet Init_X(u_X) \wedge G(u_X, u_Y)) \tag{5}$$

Thirdly, there is a relation between the operation sets of *SysX* and *SysY*, written $\geqslant_{\mathrm{Ops}_{X,Y}}$, though we suppress the subscript if the context makes it clear. Three things hold for $\geqslant_{\mathrm{Ops}_{X,Y}}$. The first of them is that $\geqslant_{\mathrm{Ops}_{X,Y}}$ is onto (i.e, is surjective): $\geqslant_{\mathrm{Ops}_{X,Y}} : \mathrm{Ops}_X \leftrightarrow\!\!\!\!\rightarrow \mathrm{Ops}_Y$.[4] The second of them is that for every case of $Op_X \geqslant Op_Y$ there are input and output relations, $In_{Op_{X,Y}} : \mathrm{I}_{Op_X} \leftrightarrow\!\!\!\!\rightarrow \mathrm{I}_{Op_Y}$ and $Out_{Op_{X,Y}} : \mathrm{O}_{Op_X} \leftrightarrow \mathrm{O}_{Op_Y}$, with the input relation onto, as indicated.[5] The third of them is that for every case of $Op_X \geqslant Op_Y$, the (forward simulation) refinement correctness PO holds:

$$\begin{aligned} &G(u_X, u_Y) \wedge In_{Op_{X,Y}}(i_X, i_Y) \wedge stp_{Op_Y}(u_Y, i_Y, u'_Y, o_Y) \Rightarrow \\ &\quad (\exists\, u'_X, o_X \bullet stp_{Op_X}(u_X, i_X, u'_X, o_X) \wedge G(u'_X, u'_Y) \wedge Out_{Op_{X,Y}}(o_X, o_Y)) \end{aligned} \tag{6}$$

The various relations introduced to instrument the refinement, namely G, $\geqslant_{\mathrm{Ops}_{X,Y}}$, and the relations $In_{Op_{X,Y}}$ and $Out_{Op_{X,Y}}$ together with all their properties, are referred to as the **refinement data**, and are written $[G/In/Out]$ for short, although this suppresses any operation dependence in the input and output relations. Below, referring to refinement data $[G/In/Out]$, but without explicitly claiming that a refinement holds, means that the data in question has all the properties stated, but does not presume that the POs (5) and (6) are true.

If we have a refinement $SysX \geqslant SysY$, and we wish to emphasise the data relevant to it, we can write $SysX \geqslant_{[G/In/Out]} SysY$.

When we have refinement data $[G/In/Out]$, and all the items mentioned in (6) are true, i.e. when we have $Op_X \geqslant Op_Y$ and:

$$\begin{aligned} &G(u_X, u_Y) \wedge In_{Op_{X,Y}}(i_X, i_Y) \wedge stp_{Op_Y}(u_Y, i_Y, u'_Y, o_Y) \wedge \\ &stp_{Op_X}(u_X, i_X, u'_X, o_X) \wedge G(u'_X, u'_Y) \wedge Out_{Op_{X,Y}}(o_X, o_Y) \end{aligned} \tag{7}$$

then we say that the steps $u_X \text{-}(i_X, Op_X, o_X)\!\!\rightarrowtail u'_X$ and $u_Y \text{-}(i_Y, Op_Y, o_Y)\!\!\rightarrowtail u'_Y$ are *in simulation*. However, when two steps are in simulation in this manner, it does not imply that a refinement $SysX \geqslant_{[G/In/Out]} SysY$ holds, unless we say so. Straightforwardly, we have (see e.g., [1, 59, 70]):

Theorem 3.1 (Trace Inclusion). *Let* $[\, u_{Y,0} \text{-}(i_{Y,0}, Op_{Y,0}, o_{Y,1})\!\!\rightarrowtail u_{Y,1} \text{-}(i_{Y,1}, Op_{Y,1}, o_{Y,2})\!\!\rightarrowtail u_{Y,2} \ \ldots\,]$ *be an arbitrary execution of SysY starting from initial state* $u_{Y,0}$*, and let* $SysX \geqslant_{[G/In/Out]} SysY$*. Then there is a corresponding execution of SysX,* $[\, u_{X,0} \text{-}(i_{X,0}, Op_{X,0}, o_{X,1})\!\!\rightarrowtail u_{X,1} \text{-}(i_{X,1}, Op_{X,1}, o_{X,2})\!\!\rightarrowtail u_{X,2} \ \ldots\,]$*, starting from an initial state* $u_{X,0}$ *guaranteed to exist by (5), in which each pair of corresponding steps* $u_{X,k} \text{-}(i_{X,k}, Op_{X,k}, o_{X,k+1})\!\!\rightarrowtail u_{X,k+1}$ *and* $u_{Y,k} \text{-}(i_{Y,k}, Op_{Y,k}, o_{Y,k+1})\!\!\rightarrowtail u_{Y,k+1}$ *are in simulation.*

Proof. This an easy induction, using (5) for initialisation and (6) for the inductive step. The assumptions that $\geqslant_{\mathrm{Ops}_{X,Y}}$ is onto Ops_Y, and that $In_{Op_{X,Y}}$ is onto I_{Op_Y}, guarantee that the hypothesis of (6) can always be satisfied for the next *SysY* step. □

The idea of trace inclusion/simulation recurs frequently in the sequel, though it is often based on concepts that go beyond simple step by step induction.

To exclude inconvenient side cases, we will henceforth assume that all systems (e.g. *SysX*) are *non-isolated*, i.e. their state space (e.g. U_X) contains no state that is not either the before-state or the after-state of some transition.

[4] We use notations (such as $\leftrightarrow$ for relations and $\leftrightarrow\!\!\!\!\rightarrow$ for surjective relations) that are familiar from the Z notation [51].

[5] Since $\geqslant_{\mathrm{Ops}_{X,Y}}$ need not be injective, we should, strictly speaking, write In_{Op_X,Op_Y} and Out_{Op_X,Op_Y}, but we do not do so unless really needed, to avoid verbosity.

Regarding the various restricted systems considered in Section 2, there is the following. If $SysX \geqslant_{[G/In/Out]} SysY$ and we also have:

$$Inv_Y(u_Y) \wedge G(u_X, u_Y) \Rightarrow Inv_X(u_X) \tag{8}$$

for invariants Inv_X on $SysX$ and Inv_Y on $SysY$, then also $SysInvX \geqslant_{[G/In/Out]} SysInvY$, and the image of $SysInvY$ through $[G/In/Out]$ is a subsystem of $SysInvX$.

Likewise, if $SysX \geqslant_{[G/In/Out]} SysY$, and there are contingent invariants $CInv_X$ and $CInv_Y$ with respect to safe input sets $CInv|_{Op_X}$ and $CInv|_{Op_Y}$, and we also have:

$$CInv_Y(u_Y) \wedge G(u_X, u_Y) \Rightarrow CInv_X(u_X) \tag{9}$$

$$i_Y \in CInv|_{Op_Y} \wedge In_{Op_{X,Y}}(i_X, i_Y) \Rightarrow i_X \in CInv|_{Op_X} \tag{10}$$

then also $SysCInvX \geqslant_{[G/In/Out]} SysCInvY$, and the image of $SysCInvY$ through $[G/In/Out]$ is a subsystem of $SysCInvX$.

The inductive proof of the trace inclusion theorem immediately implies that for each of these $\geqslant_{[G/In/Out]}$-related pairs, a trace inclusion theorem holds, and their reachable subsystems are also $\geqslant_{[G/In/Out]}$-related, i.e. $ReachSysX \geqslant_{[G/In/Out]} ReachSysY$ and $ReachSysInvX \geqslant_{[G/In/Out]} ReachSysInvY$ and $ReachSysCInvX \geqslant_{[G/In/Out]} ReachSysCInvY$, and subsystem inclusion through $[G/In/Out]$ also holds (as do further subsystem inclusions generated by composing any of the present ones with inclusions that hold within $SysX$ itself).

For $SysInvX$ and $SysInvY$, if $SysX \geqslant_{[G/In/Out]} SysY$, but (8) does not hold, we can find a stronger invariant on $SysY$, $InvG_Y$, that restricts Inv_Y to the part of $SysY$ relevant to the refinement $\geqslant_{[G/In/Out]}$, and which is given by:

$$InvG_Y(u_Y) \;\equiv\; Inv_Y(u_Y) \wedge (\exists\, u_X \bullet G(u_X, u_Y)) \tag{11}$$

With (11), the refinement correctness PO (6) guarantees that if $(Inv_Y \wedge (\exists\, u_X \bullet G))$ holds in the before-state of a $SysY$ transition, then the same holds in the after-state. This yields system $SysInvG_Y$, for which $SysX \geqslant_{[G/In/Out]} SysInvG_Y$, and subsystem inclusion through $[G/In/Out]$, both hold.

The same argument can be replayed with the even stronger $InvG_{XY}$, given by:

$$InvG_{XY}(u_Y) \;\equiv\; Inv_Y(u_Y) \wedge (\exists\, u_X \bullet G(u_X, u_Y) \wedge Inv_X(u_X)) \tag{12}$$

This yields system $SysInvG_{XY}$, for which $SysInvX \geqslant_{[G/In/Out]} SysInvG_{XY}$, and subsystem inclusion through $[G/In/Out]$, both hold.

The same approach may be applied to systems equipped with contingent invariants, if it happens that (9) and/or (10) fail to hold. In each such case, we may strengthen $CInv_Y$ by conjoining $(\exists\, u_X \bullet G)$ (or $(\exists\, u_X \bullet G \wedge CInv_X)$) to it, and/or strengthen membership of $CInv|_{Op_Y}$ by conjoining $(\exists\, i_X \bullet In_{Op_{X,Y}})$ (or $(\exists\, i_X \bullet In_{Op_{X,Y}} \wedge i_X \in i_X \in CInv|_{Op_X})$) to it. We do not write down all the details.

A further variation on these ideas is the notion of $Init$-constrained refinement. This has an additional gluing relation G_{Init} (the $Init$-constraint) which must satisfy:

$$G_{Init}(u_X, u_Y) \Rightarrow G(u_X, u_Y) \tag{13}$$

and for which the initialisation proof obligation is modified to:

$$Init_Y(u_Y) \Rightarrow (\exists\, u_X \bullet Init_X(u_X) \wedge G_{Init}(u_X, u_Y)) \tag{14}$$

while all else remains the same. Clearly, all we said just above about different variants of the basic refinement idea applies equally well to $Init$-constrained versions of them (and, of course, each $Init$-constrained variant implies its non-$Init$-constrained variant).

The kind of variations that we have introduced on the basic refinement idea prove useful later in the paper in our discussion of the case study. Furthermore, the common case in which gluing and

other relations are functional from concrete to abstract, and are total and onto, makes most of the distinctions just discussed melt away.

3.1 Generalisations

Our notion of refinement is fairly generic, in permitting an arbitrary correspondence between operation names that are related by the correctness relationship in the two systems — often a 1-1, or otherwise restricted discipline is insisted on in refinement notions. Also, many syntactically based notions of systems facilitate the definition of their transitions by incorporating guard or precondition clauses, and facilitate the definition of the (syntactically described) refinement correctness relationship by additional applicability criteria for operations that are to be related by correctness. In our case, systems are defined semantically, so the domain of the stp_{Op_X} relation does not need separate definition.

Moreover, despite possible superficial appearances to the contrary, the above framework will also suffice for the kind of continuous transitions that are needed for hybrid and cyber-physical systems. Key to this is the proviso that time is considered as a *parameter* of the dynamics and not as a normal assignable program variable. Assuming this to be so, a transition like $u_X \text{-}(i_X, Op_X, o_X)\!\!\rightarrow u'_X$ (which in a purely discrete transition system would take place at some index k of an execution, e.g. $u_{X,k} \text{-}(i_{X,k}, Op_{X,k}, o_{X,k+1})\!\!\rightarrow u_{X,k+1}$), instead becomes a family of before/after pairs indexed by a left-closed right-open interval of time $t \in [\mathbb{t}_\mathrm{L} \ldots \mathbb{t}_\mathrm{R})$, with the before-state fixed at $\mathbb{t}_\mathrm{L}$ and the after-state ranging over the open interval $t \in (\mathbb{t}_\mathrm{L} \ldots \mathbb{t}_\mathrm{R})$ thus: $u_X(\mathbb{t}_\mathrm{L}) \text{-}(i_{[\mathbb{t}_\mathrm{L} \ldots t]}, Op_{X,k}, o_{[\mathbb{t}_\mathrm{L} \ldots t]})\!\!\rightarrow u_X(t)$. In this, for each value of $t \in (\mathbb{t}_\mathrm{L} \ldots \mathbb{t}_\mathrm{R})$, the input i and output o are functions of time defined over the interval $[\mathbb{t}_\mathrm{L} \ldots t)$. Continuous behaviour, and its relationship to the present framework is discussed in more detail in Section 5.

A further generalisation of the framework described is to allow the individual transitions of *SysX* and/or *SysY* that appear in (6) to be replaced by sequences of transitions of length >1. In the purely discrete case this generates so-called (m, n) correctness diagrams (as used in the ASM formalism's refinement notion [26, 27]). In this generalisation, to establish correctness, the (m, n) diagrams need to abut along an execution in the same way that the 1-1 simulation squares (7) abut in the trace refinement discussed above. In the time parameterised hybrid/cyber-physical case, the same approach works the most easily if we insist that the *SysX* and *SysY* transition sequences in a PO like (6) start and end with transitions of the same kind,[6] even if this is not strictly necessary. The (m, n) diagram concept will be discussed at greater length in Section 6.2, and will be needed in the case study.

3.2 Composition

In the development of complex systems by refinement, breaking up the development into more than one refinement step is very helpful.[7] This makes the composition of refinement steps of interest. Vertical composition of refinement steps arises when we have three systems, *SysX*, *SysY*, *SysZ*, and $SysX \geqslant SysY$ and $SysY \geqslant SysZ$. Then we get a generic refinement $SysX \geqslant SysZ$, given by data:

$$\geqslant_{\mathrm{Ops}_{X,Z}} \equiv \geqslant_{\mathrm{Ops}_{X,Y}} \fatsemi \geqslant_{\mathrm{Ops}_{Y,Z}} \tag{15}$$

$$G_{X,Z} \equiv G_{X,Y} \fatsemi G_{Y,Z} \tag{16}$$

$$In_{Op_{X,Z}} \equiv In_{Op_{X,Y}} \fatsemi In_{Op_{Y,Z}} \tag{17}$$

$$Out_{Op_{X,Z}} \equiv Out_{Op_{X,Y}} \fatsemi Out_{Op_{Y,Z}} \tag{18}$$

[6] "The same kind" refers to the *SysX* and *SysY* transition sequences both starting with a discrete transition, or both starting with a time parameterised transition — similarly for the transitions at the ends of the two sequences.

[7] This view has been particularly promoted in the B-Method family of methodologies [2, 3].

In the above $\fatsemi$ is the conventional sequential composition of relations, which is obviously transitive. And since refinement data is just a triple of relations, transitivity of refinement follows. The onto assumptions in $SysX \geqslant SysY$ and $SysY \geqslant SysZ$ compose smoothly, and this enables trace inclusion and all other properties mentioned above to follow readily by induction using the composition/decomposition of the component relations defined in (15)-(18). In [14, 15, 21] a wide variety of other composition mechanisms are discussed in detail for retrenchment, covered in the next section. These readily reduce to analogous results for the refinement notion given here. Since they are not of great importance in the rest of the paper, we do not discuss them further.

4 RETRENCHMENT

As noted earlier, retrenchment [14, 15, 21, 66] was introduced to tackle development stages that did not naturally and comfortably fit into known formal refinement approaches. Observing how successful refinement had been in capturing many developments in a rigorous way, the aspiration was to develop a framework which 'had a similar shape' (i.e. a framework of formal PO schemas that are instantiated from a specific pair of system models and relations between them) but allowed greater flexibility in what these contained, and what was claimed. The aim of this was to allow the refinement and retrenchment ideas to coexist harmoniously when this was possible.

Being a weaker relationship between systems than refinement, naturally the guarantees that refinement can offer (e.g. preserving certain properties proved at a higher level of abstraction to lower levels) are forfeit, but at least it gives the capability to express *some* relationship between the systems rather than none at all. Our ambition in this paper is to push out the boundaries of the approach further than hitherto, while simultaneously simplifying and streamlining the notion in the light of experience.

Given the framework for systems given in the previous section, retrenchment is easy to describe. Thus, let $SysX$ and $SysY$ be two systems as above. We will assume that $SysX$ is provided with an invariant Inv_X, for which the POs (1) and (2) hold; analogously for $SysY$. These invariants play little part in the theory developed below, but it is convenient to assume that they are present for discussing the comparison with refinement. We say that $SysX$ is *retrenched* by $SysY$, written $SysX \succcurlyeq SysY$, provided we have the following.

Firstly, there is a gluing relation between the state spaces $G : \mathsf{U}_X \leftrightarrow \mathsf{U}_Y$. Note though, that G plays a rather different role here than in the context of refinement. We stipulate that the initialisation PO (5) holds.

Secondly, we insist that there is a relation between the operation sets of $SysX$ and $SysY$, written $\succcurlyeq_{\mathrm{Ops}_{X,Y}}$, again suppressing the subscript if it is convenient, for which two things hold. The first is that, for every case of $Op_X \succcurlyeq Op_Y$ there is a within relation $W_{Op_{X,Y}} : \mathsf{U}_X \times \mathsf{I}_{Op_X} \leftrightarrow \mathsf{U}_Y \times \mathsf{I}_{Op_Y}$ and a delivers relation $D_{Op_{X,Y}} : \mathsf{U}_X \times \mathsf{I}_{Op_X} \times \mathsf{U}_X \times \mathsf{O}_{Op_X} \leftrightarrow \mathsf{U}_Y \times \mathsf{I}_{Op_Y} \times \mathsf{U}_Y \times \mathsf{O}_{Op_Y}$.[8] The second is that, whenever $Op_X \succcurlyeq Op_Y$, the retrenchment correctness PO holds:

$$\begin{aligned} &W_{Op_{X,Y}}(u_X, i_X, u_Y, i_Y) \wedge stp_{Op_Y}(u_Y, i_Y, u'_Y, o_Y) \Rightarrow \\ &\quad (\exists\, u'_X, o_X \bullet stp_{Op_X}(u_X, i_X, u'_X, o_X) \wedge D_{Op_{X,Y}}(u_X, i_X, u'_X, o_X, u_Y, i_Y, u'_Y, o_Y)) \end{aligned} \tag{19}$$

As for refinement, the relations introduced to instrument the retrenchment, namely G, $\succcurlyeq_{\mathrm{Ops}_{X,Y}}$, and the relations $W_{Op_{X,Y}}$ and $D_{Op_{X,Y}}$ together with all their properties, are referred to as the **retrenchment data**, and are written $[G/W/D]$ for short. Below, we refer to retrenchment data $[G/W/D]$, without claiming that the POs (5) and (19) are true, as needed.

[8] Since $\succcurlyeq_{\mathrm{Ops}_{X,Y}}$ need not be injective, we should, strictly speaking, write W_{Op_X,Op_Y} and D_{Op_X,Op_Y}, but we do not do so to avoid verbosity.

If we have a retrenchment $SysX \succcurlyeq SysY$, and we wish to emphasise the data relevant to it, we can write $SysX \succcurlyeq_{[G/W/D]} SysY$.

When we have retrenchment data $[G/W/D]$, and all the items mentioned in (19) are true, i.e. when we have $Op_X \succcurlyeq Op_Y$ and:

$$W_{Op_{X,Y}}(u_X, i_X, u_Y, i_Y) \wedge stp_{Op_Y}(u_Y, i_Y, u'_Y, o_Y) \wedge$$
$$stp_{Op_X}(u_X, i_X, u'_X, o_X) \wedge D_{Op_{X,Y}}(u_X, i_X, u'_X, o_X, u_Y, i_Y, u'_Y, o_Y) \tag{20}$$

then we say that the steps u_X -(i_X, Op_X, o_X)-$\succ$ u'_X and u_Y -(i_Y, Op_Y, o_Y)-$\succ$ u'_Y are *in simulation*. However, when two steps are in simulation in this manner, it does not imply that a retrenchment $SysX \succcurlyeq_{[G/W/D]} SysY$ holds, unless we say so.

If $SysX \succcurlyeq_{[G/W/D]} SysY$ and we are given $stp_{Op_Y}(u_Y, i_Y, u'_Y, o_Y)$ (and u_X), and can then find other data items to make (20) true, then we say that $stp_{Op_Y}(u_Y, i_Y, u'_Y, o_Y)$ is simulable (from u_X).

The evident structural similarity between our formulations of refinement and retrenchment indicates a couple of things. For one, if we have a refinement $SysX \succcurlyeq_{[G/In/Out]} SysY$ and we set

$$W_{Op_{X,Y}}(u_X, i_X, u_Y, i_Y) \;\equiv\; G(u_X, u_Y) \wedge In_{Op_{X,Y}}(i_X, i_Y) \tag{21}$$

and

$$D_{Op_{X,Y}}(u_X, i_X, u'_X, o_X, u_Y, i_Y, u'_Y, o_Y) \;\equiv$$
$$G(u_X, u_Y) \wedge In_{Op_{X,Y}}(i_X, i_Y) \wedge G(u'_X, u'_Y) \wedge Out_{Op_{X,Y}}(o_X, o_Y) \tag{22}$$

then refinement immediately becomes a special case of retrenchment. For another, the various generalisations of the basic refinement notion discussed at the end of Section 3 carry over immediately to the retrenchment sphere.

In the case of refinement, the assumption that the refinement correctness PO (6) holds for $SysX$ and $SysY$ strengthens the facts that the reachable sets of the two systems each satisfy their respective invariants Inv_X and Inv_Y, and is connected with the fact that $InvG_Y$ may be properly stronger than Inv_Y. In the general case of retrenchment, there is no relationship between W and D on the one hand, and G on the other. So although the reachable sets of the two systems each satisfy their respective invariants, no trace inclusion conclusion can be drawn on the basis of the retrenchment PO itself, and hence no relationship between the reachable sets can be deduced.

For a retrenchment $SysX \succcurlyeq_{[G/W/D]} SysY$, if we have that the $\succcurlyeq_{\mathrm{Ops}_{X,Y}}$ relation is onto, $\succcurlyeq_{\mathrm{Ops}_{X,Y}}$: $\mathrm{Ops}_X \leftrightarrow\!\!\!\!\rightarrow \mathrm{Ops}_Y$, and that for each case of $Op_X \succcurlyeq Op_Y$, the within relation is onto, $W_{Op_{X,Y}}$: $\mathrm{U}_X \times \mathrm{I}_{Op_X} \leftrightarrow\!\!\!\!\rightarrow \mathrm{U}_Y \times \mathrm{I}_{Op_Y}$, then we say that the retrenchment is *onto*. The same applies if we merely have retrenchment data $[G/W/D]$, whereupon the we speak of *onto retrenchment data*. Refinements and onto retrenchments are the basis of a similar range of generic simulation results investigated in Section 8.

Note that the onto property alone does not lead to a trace inclusion theorem based on an inductive argument, since the truth of D for one pair of steps does not guarantee the truth of W for the next pair. Trace inclusion thus demands stronger properties of the W or D relations than are available generically. This does not mean that trace inclusion is not a desirable goal, but it does mean that to the extent that it might be true in any particular application of retrenchment, it would have to be pursued by more bespoke means, rather than by simply applying a generic inductive strategy. (A desirable situation might be where the incorporation of a small amount of bespoke reasoning can fix a relationship between models, the remainder of which can be handled using refinement.)

4.1 Composition of Retrenchments

The simpler retrenchment structure that we have defined here makes the vertical composition of our retrenchments directly analogous to (15), (17), (18), with the replacements W/In and D/Out,

and taking the different signatures into account. Noting that the absence of the onto assumptions does not affect the well-definedness of the relational compositions in (15), (17), (18), confirms that this composition law works. We can observe that this yields a slightly simpler composition story for retrenchments than appears in [14, 15].

The same ideas yield a method for composing retrenchments with refinements (and vice versa). We simply interpret each refinement in such a composition as a retrenchment in the way indicated above, and then compose the resulting retrenchments as just described, the result being a retrenchment in every case.

When we have two systems X and Y and more than one retrenchment can be shown between them, e.g. $SysX \succcurlyeq SysY$ with data $[G/W_1/D_1]$ and $[G/W_2/D_2]$ respectively, these retrenchments can be composed in various ways. Thus, retrenchment data $[G/W_1/D_1]$ and $[G/W_2/D_2]$ may be combined to give $[G/W_1 \vee W_2/D_1 \vee D_2]$, or, more incisively, $[G/W_1 \vee W_2/(W_1 \Rightarrow D_1) \wedge (W_2 \Rightarrow D_2)]$ (or even $[G/W_1 \vee W_2/(W_1 \wedge D_1) \vee (W_2 \wedge D_2)]$ if we have sufficiently strong completeness properties concerning the models involved). Possibilities based on conjunction exist too, of course. The corresponding retrenchments can be shown to follow by generic arguments, and evidently also apply to refinements. Thoughts like these give rise to many possibilities for adapting the theory developed in this paper more closely to particular applications.

4.2 Earlier Retrenchment Notions

For clarity, it is worth noting that the notion of retrenchment presented above differs in detail from the one in [14, 15, 21, 66] and elsewhere. The key difference is that in the earlier work, the retrenchment correctness PO (19) appears as:

$$G \wedge P_{Op_{X,Y}} \wedge stp_{Op_Y} \Rightarrow ((G' \wedge O_{Op_{X,Y}}) \vee C_{Op_{X,Y}}) \tag{23}$$

In (23), G is a gluing relation as in Section 3; also P has the signature of W here, and O and C both have the signature of D here.

The explanation for the simpler present structure comes from past experience. It was often found in practice that G was not needed in applications of the retrenchment notion, so became defaulted to true. (In the present formulation, although G is present, it plays a lesser role, being mainly useful for gauging the closeness to, and/or deviation from, refinement.)

Additionally, the top level propositional structure of the conclusion of (23) is less critical in the present formulation, and will prove to be so in our motivating case study, so we suppress it here (the required propositional structure can always be re-imposed in the internal shape of D if necessary).

5 TAKING CONTINUOUS BEHAVIOUR INTO ACCOUNT

All of the above (and a good deal of what is to come below) has been expressed in the notations usually used for discussing conventional discrete transition systems. Apparently, this would make no allowance for the requirements of hybrid and cyber-physical systems, for which, continuous state change is an unavoidable ingredient. However, this is not so. There is a systematic way of reinterpreting any formula that is needed for the discrete systems case, into a formula appropriate for the corresponding continuous behaviour case.

This comes about because in the physical or engineering theory literature, real world time is treated as a *parameter*, and *not* as a dynamical variable of the system. This enables us to talk about values of time, about intervals of time, and about values of variables at different times, but not about updates of time, as such. The latter are thus implemented by your favourite deity (rather than by the system in question), and are dealt with (implicitly in the usual physical or engineering theory literature) by quantifying formulas involving time-varying variables over ranges of the time parameter. Consequently, the reinterpretation needed amounts to manipulation of parameters.

We argue as follows. When a purely discrete system is interpreted in the real world, each state encountered during an execution persists for some interval of time, to be duly succeeded by the next state in the execution. If we interpret time as a semi-infinite portion of the reals $[t_0 \ldots \infty) \subseteq \mathbb{R}$, then these intervals can be modelled as a succession of left-closed, right-open subintervals, $[t_0 \ldots \infty) \equiv [t_0 \ldots t_1) \cup [t_1 \ldots t_2) \cup \ldots$. Evidently, such subintervals can conveniently abut without gaps or overlaps.

The succession of values of a discrete state variable now become piecewise constant functions of time, constant over these subintervals. The join points of these intervals, e.g. t_1 at the join of $[t_0 \ldots t_1)$ and $[t_1 \ldots t_2)$, allow for the discontinuous changes required by the discrete transition parts of a system model. For such a transition, taking place at t_1, the before-value of a variable x say, would correspond to the limiting value at t_1 from smaller values in the subinterval $[t_0 \ldots t_1)$ (of the relevant function of time), and the after-value x' would correspond to the initial value in the subinterval $[t_1 \ldots t_2)$ (being also the limiting value from bigger values in $[t_1 \ldots t_2)$) (of the same function of time).

The same approach allows portions of continuously varying behaviour in a system to be modelled as continuously changing functions whose domains are also such left-closed, right-open subintervals of the time parameter.[9] Altogether, this fixes the semantic framework in which both discrete and continuously varying behaviour can be interpreted in a unified manner.

We now consider the translation of the formulas of the preceding sections into the world we have just constructed. First of all, the discrete formulas remain unchanged, though interpreted as just described regarding before-values and after-values, when required for the discrete transitions taking place at the join points of successive left-closed, right-open subintervals. The same applies to formulas concerning initial states, though there is no predecessor subinterval then.

For formula analogues needed for the continuously varying behaviours in the interior of a subinterval $[\mathbb{t}_\mathrm{L} \ldots \mathbb{t}_\mathrm{R})$, (using $\mathbb{t}_\mathrm{L}$ and $\mathbb{t}_\mathrm{R}$ as generic notations for the endpoints), we first assume that the original discrete formula in question Φ, has the propositional structure $\Phi \equiv \mathit{HYP} \Longrightarrow \mathit{CONC}$ at top level. A formula of this shape will be required to hold throughout $[\mathbb{t}_\mathrm{L} \ldots \mathbb{t}_\mathrm{R})$. The basic inspiration underpinning the translation is the idea that a before-value in Φ is tied to the corresponding value at $\mathbb{t}_\mathrm{L}$, and the corresponding after-value in Φ maps to the family of values in the open interval $(\mathbb{t}_\mathrm{L} \ldots \mathbb{t}_\mathrm{R})$. However, if no corresponding after-value occurs in Φ, then the before-value maps to the family of values in the open interval $(\mathbb{t}_\mathrm{L} \ldots \mathbb{t}_\mathrm{R})$ too.[10]

Now, Φ will contain a number of logical atoms, combined using logical connectives. Some atoms are used just as hypotheses in *HYP*, e.g. inputs and within relations. Since such atoms depend only on before-values in the discrete case, there are no after-values present in Φ (and indeed the hypotheses *HYP* must hold throughout the interval). Therefore our recipe stipulates that the variables in these atoms become functions of time, thus: $\mathit{In}_{Op_{X,Y}}(i_X(t), i_Y(t))$ and $W_{Op_{X,Y}}(u_X(t), i_X(t), u_Y(t), i_Y(t))$.

Other atoms are used just as conclusions in *CONC*, e.g. outputs. Their variables therefore become functions of time, thus: $\mathit{Out}_{Op_{X,Y}}(o_X(t), o_Y(t))$, and $\mathit{Out}_{Op_{X,Y}}$ must hold throughout the interval.

Invariants (and similar relations restricted to only relating states to one another) are normally assumed in *HYP* and reestablished in *CONC*. So, according to our recipe, the occurrence in *HYP* should correspond to the value at $\mathbb{t}_\mathrm{L}$, and the occurrence in *CONC* will correspond to the family of values in the open interval $(\mathbb{t}_\mathrm{L} \ldots \mathbb{t}_\mathrm{R})$.

The *stp* relations contain before- and after- state values, and inputs and outputs, so all the preceding apply. An instance such as $\mathit{stp}_{Op_X}(u_X, i_X, u'_X, o_X)$ becomes $\mathit{stp}_{Op_X}(u_X(\mathbb{t}_\mathrm{L}), i_X(t), u_X(t), o_X(t))$,

[9] Since the functions of time are no longer piecewise constant, it is the responsibility of the semantics of any formalism using the concepts of this paper to ensure that the required limits actually exist.

[10] The value at $\mathbb{t}_\mathrm{L}$ must agree with the limit value from the right.

where t ranges over the relevant interval. Delivers relations are similar to stp, potentially containing all the preceding items. An instance such as $D_{Op_{X,Y}}(u_X, i_X, u'_X, o_X, u_Y, i_Y, u'_Y, o_Y)$ therefore becomes $D_{Op_{X,Y}}(u_X(\mathbb{t}_L), i_X(t), u_X(t), o_X(t), u_Y(\mathbb{t}_L), i_Y(t), u_Y(t), o_Y(t))$.[11]

Proceeding beyond individual atoms, which are translated in a syntactically driven manner, the various occurrences of time in the atoms in a formula like $\Phi \equiv HYP \Rightarrow CONC$ must be properly correlated. This is accomplished by a single, outer level quantification over the requisite time interval (since physical time proceeds at the same rate shared by all physical systems and quantities (in classical physics)). Thus, the formal discrete to continuous transformation schema can be expressed as:

$$\Psi_1(x) \ldots \Psi_2(x, x') \quad \longrightarrow\!\!\triangleright \quad \big(\forall t \in (\mathbb{t}_L \ldots \mathbb{t}_R) \bullet \Psi_1(x(t)) \ldots \Psi_2(x(\mathbb{t}_L), x(t)) \big) \tag{24}$$

In (24), $\Psi_1(x)$ and $\Psi_2(x, x')$ are individual atoms occurring in Φ, the ellipsis indicating other parts of Φ. It is important to note that (24) remains unchanged whether we know in advance or not (on whatever basis), what the values of $\mathbb{t}_L$ and $\mathbb{t}_R$ are in any given case. Such knowledge affects *how we infer conclusions* from (24), but not the form of (24) itself. Thus, if we know $\mathbb{t}_L$ and $\mathbb{t}_R$, we can substitute them directly, and deduce some consequences. But very often, in an actual execution of the system, there are many occurrences of the continuous behaviour specified by the same syntactic fragment, so the values of $\mathbb{t}_L$ and $\mathbb{t}_R$ for any individual occurrence depend on the rest of the system and are thus global properties. So deducing analogous consequences becomes a different challenge.

We give some examples of earlier system properties translated so that they apply to continuously changing behaviour. First, we show how the maintenance of invariants PO (2) fares under our scheme. This translates to:

$$\big(\forall t \in (\mathbb{t}_L \ldots \mathbb{t}_R) \bullet Inv_X(u_X(\mathbb{t}_L)) \wedge stp_{Op_X}(u_X(\mathbb{t}_L), i_X(t), u_X(t), o_X(t)) \Rightarrow Inv_X(u_X(t)) \big) \tag{25}$$

Second, we treat the refinement correctness PO (6). This translates to:

$$\begin{aligned} &\big(\forall t \in (\mathbb{t}_L \ldots \mathbb{t}_R) \bullet \\ &\quad G(u_X(\mathbb{t}_L), u_Y(\mathbb{t}_L)) \wedge In_{Op_{X,Y}}(i_X(t), i_Y(t)) \wedge stp_{Op_Y}(u_Y(\mathbb{t}_L), i_Y(t), u_Y(t), o_Y(t)) \Rightarrow \\ &\qquad (\exists\, u_X(t), o_X(t) \bullet stp_{Op_X}(u_X(\mathbb{t}_L), i_X(t), u_X(t), o_X(t)) \wedge \\ &\qquad\qquad G(u_X(t), u_Y(t)) \wedge Out_{Op_{X,Y}}(o_X(t), o_Y(t))) \big) \end{aligned} \tag{26}$$

Likewise the retrenchment correctness PO (19) —bearing in mind footnote 11— becomes:

$$\begin{aligned} &\big(\forall t \in (\mathbb{t}_L \ldots \mathbb{t}_R) \bullet \\ &\quad W_{Op_{X,Y}}(u_X(t), i_X(t), u_Y(t), i_Y(t)) \wedge stp_{Op_Y}(u_Y(\mathbb{t}_L), i_Y(t), u_Y(t), o_Y(t)) \Rightarrow \\ &\qquad (\exists\, u_X(t), o_X(t) \bullet stp_{Op_X}(u_X(\mathbb{t}_L), i_X(t), u_X(t), o_X(t)) \wedge \\ &\qquad\qquad D_{Op_{X,Y}}(u_X(\mathbb{t}_L), i_X(t), u_X(t), o_X(t), u_Y(\mathbb{t}_L), i_Y(t), u_Y(t), o_Y(t)) \big) \end{aligned} \tag{27}$$

[11] The policy just described has consequences to be aware of. Whereas for a discrete transition, typical before- and after-state values u and u' refer to different limits, for a continuously varying transition, a value $u(\mathbb{t}_L)$ refers to the initial value of u in $[\mathbb{t}_L \ldots \mathbb{t}_R)$, i.e. the limit from the right at $\mathbb{t}_L$. It is the responsibility of the semantics of any formalism using the concepts of this paper to ensure that this approach is globally consistent. This is particularly incisive in the case of the within relation W, which, in the discrete case, contains both before-state values and inputs. In the continuous case, the inputs have duration (as do the state values of course), but state values are typically *to be determined by the dynamics*, and not merely assumed in the hypotheses. It is the responsibility of the semantics of any formalism using the concepts of this paper to have a clear policy on this point. An obvious policy is to restrict W to state values at $\mathbb{t}_L$ only, allowing the inputs to have their duration. An obvious alternative is to regard any assumption in W about state values during a continuous transition to be interpreted as a *constraint* on the dynamics, ensuring that inconsistent specifications of dynamics are suitably handled. This issue can become yet more complicated when nontrivial (m, n) diagrams, discussed in Section 6, are allowed.

From the above, the modification of properties with a shape different from $\Phi \equiv HYP \Longrightarrow CONC$, such as simulation properties, e.g. (7) or (20), follows easily. With the above understood, we continue our convention of using discrete systems notations, as convenient, below.

6 SIMULATION NOTIONS FOR RETRENCHMENTS AND THEIR DATA

We noted when discussing the notion of simulation for refinement (7), the truth of the simulation notion for two steps u_X -(i_X, Op_X, o_X)-> u'_X and u_Y -(i_Y, Op_Y, o_Y)-> u'_Y requires only the presence of refinement data, and does not imply that there is an actual refinement $SysX \geqslant_{[G/In/Out]} SysY$ from $SysX$ to $SysY$. The latter is a strengthening of the assumptions needed for the former. In this section, we explore a number of notions related to simulation in the retrenchment context. We will set these out assuming just retrenchment data. They will be equally applicable when there is an actual retrenchment, just as for refinement. Moreover, all such notions have $Init$-constrained versions, if we strengthen the requirement demanded of the initial states of $SysX$ and $SysY$. Various of these notions prove useful below.

6.1 Simulation Notions for Individual Pairs of Steps

Suppose that we have retrenchment data $[G/W/D]$ from $SysX$ to $SysY$, and that a pair of steps of the two systems is in simulation, i.e. that (20) holds. If, in addition to (20), we have $G(u_X, u_Y)$ but not $G(u'_X, u'_Y)$, we say that the pair of steps is *conceding*. If, in addition to (20), we have $G(u'_X, u'_Y)$ but not $G(u_X, u_Y)$, we say that the pair of steps is *restoring*. If, in addition to (20), we have $G(u_X, u_Y)$ and also $G(u'_X, u'_Y)$, we say that the pair of steps is *refining*. Evidently, the conceding and restoring step pairs capture the transitions between the ideal refining behaviour between the two systems on the one hand, and the less desirable but often unavoidable non-refining behaviour between them on the other.

Suppose that we have retrenchment data $[G/W/D]$ from $SysX$ to $SysY$. We say that $SysY$ is *refining simulable* by $SysX$ if a trace inclusion property between $SysX$ and $SysY$, mediated by the data W/D (instead of by data $G/In/Out$ as in the refinement case) holds, and additionally, each corresponding pair of steps is refining. This means that (20) holds for each corresponding pair of abstract and concrete steps, and consecutive concrete steps (joined by a common state) are simulated by consecutive abstract steps (also joined by a common state), and for each pair of corresponding abstract and concrete states, $u_{X,k}$ and $u_{Y,k}$, $G(u_{X,k}, u_{Y,k})$ holds. It is evident that if $SysY$ is refining simulable by $SysX$, then we can construct a *bona fide* refinement trace simulation from $ReachSysX$ to $ReachSysY$ by constructing relations $In_{Op_{X,Y}}$ and $Out_{Op_{X,Y}}$ to work alongside G as follows:

$$In_{Op_{X,Y}}(i_X, i_Y) \;\equiv\; (\exists\, u_X \in \mathsf{U}_X, u_Y \in \mathsf{U}_Y \bullet W_{Op_{X,Y}}(u_X, i_X, u_Y, i_Y)) \tag{28}$$

and

$$\begin{aligned} Out_{Op_{X,Y}}(o_X, o_Y) \;\equiv\; (\exists\, u_X \in \mathsf{U}_X, i_X \in \mathsf{In}_{Op_X}, u'_X \in \mathsf{U}_X, u_Y \in \mathsf{U}_Y, i_Y \in \mathsf{In}_{Op_Y}, u'_Y \in \mathsf{U}_Y \bullet \\ D_{Op_{X,Y}}(u_X, i_X, u'_X, o_X, u_Y, i_Y, u'_Y, o_Y)) \end{aligned} \tag{29}$$

Note that we are not claiming thereby that the refinement correctness PO (6) holds; rather that the refinement simulation property (7) holds for each pair of abstract and concrete steps, and consecutive concrete steps are simulated by consecutive abstract steps (both joined by common states).

We say that $SysY$ is *strongly comprehensively simulable* by $SysX$ if a trace inclusion property between $SysX$ and $SysY$, mediated by the data W/D holds (but the stronger refining conditions involving $G(u_{X,k}, u_{Y,k})$ etc., are not asserted). We say that $SysY$ is *partially simulable over* $\mathbf{M}$ by $SysX$, where $\mathbf{M}$ is a set of executions of $SysY$, when for each execution in $\mathbf{M}$, at least some subsequence

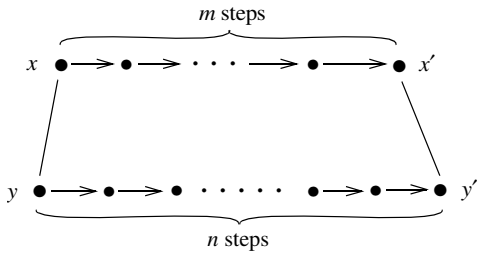

Fig. 1. An (m, n) diagram with m abstract steps and n concrete ones. The vertical relationships between x and y, and between x' and y' are unlabelled, as in this paper, different properties are appropriate according to context.

or subsequences of it is/are simulable. Since, any time any execution of *SysY* contains even a single step which is in simulation with some step of *SysX*, the definition of partial simulability is satisfied, we see that partial simulability, by itself, is a vastly weaker property than the preceding two notions. We can make it more precise by saying 'partially simulable over **M**, where Φ'. In this, Φ quantifies the extent to which executions in **M** are simulable or not in some appropriate manner.

If *SysY* is partially simulable over **M** by *SysX*, and furthermore, for each execution in **M**, any two consecutive maximal simulable subsequences (separated by one or more non-simulable steps of *SysY*) e.g.

$$[\ \ldots\ u_{Y,j}\ \text{-}(i_{Y,j}, Op_{Y,j}, o_{Y,j+1})\!\!\succ\ u_{Y,j+1}\ \text{-}(i_{Y,j+1}, Op_{Y,j+1}, o_{Y,j+2})\!\!\succ\ u_{Y,j+2}\ \ldots$$
$$\ldots\ u_{Y,k}\ \text{-}(i_{Y,k}, Op_{Y,k}, o_{Y,k+1})\!\!\succ\ u_{Y,k+1}\ \text{-}(i_{Y,k+1}, Op_{Y,k+1}, o_{Y,k+2})\!\!\succ\ u_{Y,k+2}\ \ldots\ ]$$

(where $Op_{Y,j}$ and $Op_{Y,k+1}$ are simulable (by $u_{X,\tilde{j}}\ \text{-}(i_{X,\tilde{j}}, Op_{X,\tilde{j}}, o_{X,\tilde{j}+1})\!\!\succ\ u_{X,\tilde{j}+1}$ and $u_{X,\tilde{k}+1}\ \text{-}(i_{X,\tilde{k}+1}, Op_{X,\tilde{k}+1}, o_{X,\tilde{k}+2})\!\!\succ\ u_{X,\tilde{k}+2}$ respectively, for example), but $Op_{Y,j+1}$ all the way up to $Op_{Y,k}$ (inclusive) are not),

it is nevertheless the case that, from the last simulating state in X of the first subsequence, e.g. $u_{X,\tilde{j}+1}$, to the first simulating state in X of the second subsequence, e.g. $u_{X,\tilde{k}+1}$, there is a sequence of steps of *SysX* (not necessarily of zero length even if $u_{X,\tilde{j}+1} = u_{X,\tilde{k}+1}$), then we say that *SysY* is *weakly simulable over* **M** by *SysX*. We call the sequence of steps of *SysX* in question a *bridging sequence*. If it happens that **M** is all the executions of *SysY*, then we say that *SysY* is *weakly comprehensively simulable* by *SysX*. If, in the preceding, we insist on refinement simulability for the pairs of steps that are required to be in simulation, we get the notions of *weak refinement simulability over* **M** and *weak comprehensive refinement simulability*.

In the above, the partially simulable case is appropriate when the concrete system is capable of exhibiting behaviour that is, by design, not considered at all at the abstract level. Then, when the concrete system enters such behaviour, the possibility of simulation ceases.

The weakly simulable case is appropriate when the concrete system is capable of behaviour that is not allowed for in the abstract system, but where such behaviour could nevertheless be compensated for by the abstract system engaging in unrelated behaviour that would bring the two systems back to a simulable relationship eventually.

6.2 (m, n) Diagrams, and Liberalised Refinement and Retrenchment

The idea of bridging sequences, and the possibility of averting one's gaze from the interior of a bridging sequence (and of the (non-simulable) sequence it bridges over) raises the possibility of viewing refinement, retrenchment and simulation at a coarser level of granularity. Such possibilities were used with profit in the ASM approach [27], and were distilled into the notion of (m, n) *diagram*. At its basis, an (m, n) diagram is just a pair of execution fragments, one abstract and one concrete. Fig. 1 gives an example, with m abstract steps and n concrete ones.

In ASM, concerned as it is, exclusively with refinement between discrete systems, the gluing relation is required to hold between x and y and between x' and y' (and is not expected to hold between any of the interior states of the two fragments).

In this paper, we use the same name, but apply it to a more flexible range of concepts. Moreover, Fig. 1 is deliberately imprecise about whether the transitions are just discrete changes of state, or if smooth state evolution is included. In the former case, the blobs can represent the states and the arrows the discrete transitions, while in the latter case, the arrows can represent the smooth state evolution and the blobs can represent the join points where discrete state change can occur.

Thus, focusing on discrete systems, in the weakly simulable case, consider the non-simulable concrete fragment

$$[\ u_{Y,j+1} \text{-}(i_{Y,j+1}, Op_{Y,j+1}, o_{Y,j+2})\!\!\rightarrow u_{Y,j+2} \ldots u_{Y,k} \text{-}(i_{Y,k}, Op_{Y,k}, o_{Y,k+1})\!\!\rightarrow u_{Y,k+1}\ ] \tag{30}$$

and the corresponding abstract fragment

$$[\ u_{X,\tilde{j}+1} \text{-}(i_{X,\tilde{j}+1}, Op_{X,\tilde{j}+1}, o_{X,\tilde{j}+2})\!\!\rightarrow u_{X,\tilde{j}+2} \ldots u_{X,\tilde{k}} \text{-}(i_{X,\tilde{k}}, Op_{X,\tilde{k}}, o_{X,\tilde{k}+1})\!\!\rightarrow u_{X,\tilde{k}+1}\ ] \tag{31}$$

(where no relationship between either j and $\tilde{j}$ or k and $\tilde{k}$ is implied). We call this a *weak* (m, n) *diagram*. A weak (m, n) diagram can be preceded by a simulating pair of steps and can be followed by a simulating pair of steps. (And because we are contemplating retrenchment data, this implies that in the former case D holds, while in the latter case W holds.)

If, in a weak (m, n) diagram, $G(u_{X,\tilde{j}+1}, u_{Y,j+1})$ and $G(u_{X,\tilde{k}+1}, u_{Y,k+1})$ both hold, then we call it an (m, n) *diagram*. And if, in an (m, n) diagram, no abstract state occurring in the abstract fragment is related by G to any concrete state occurring in the concrete fragment (aside from those at the extreme ends of the two fragments, as required by the definition), then we have a *strong* (m, n) *diagram*.

If, in any of the cases of: a weak (m, n) diagram, an (m, n) diagram, or a strong (m, n) diagram, it is true that $k - j = L = \tilde{k} - \tilde{j}$, then we call the diagram L*-aligned*, and, instantiating the meta variables m, n in the terminology, it becomes a (weak, _, strong) (L, L)-diagram.

The various shades of (m, n) diagram we have been discussing encourage us to liberalise the single-step-to-single-step notions of refinement and retrenchment that we originally introduced, to include analogous fragment-to-fragment notions. Given what we have said already, these are easy to define. Thus, relations $\geqslant_{\mathrm{Ops}_{X,Y}} : \mathrm{Ops}_X \leftrightarrows \mathrm{Ops}_Y$ and $\succcurlyeq_{\mathrm{Ops}_{X,Y}} : \mathrm{Ops}_X \leftrightarrow \mathrm{Ops}_Y$ change signature to $\geqslant_{\mathrm{Ops}_{X,Y}} : \mathrm{Ops}_X^* \leftrightarrow \mathrm{Ops}_Y^*$ and $\succcurlyeq_{\mathrm{Ops}_{X,Y}} : \mathrm{Ops}_X^* \leftrightarrow \mathrm{Ops}_Y^*$, where there must be sufficiently many operation sequences in the domains and ranges of these relations to give the level of coverage needed. The constituents of the refinement and retrenchment correctness POs (6) and (19), and of associated notions, now adapt smoothly to refer to execution fragments rather than single steps. In the refinement case, note that neither $\geqslant_{\mathrm{Ops}_{X,Y}}$ nor input relations, are onto. The loss of onto-ness must be compensated by additional, system dependent reasoning, to ensure adequate coverage of all concrete execution sequences (the simulation of which, we regard as the touchstone of refinement). With this in place, we can generalise each of the many notions of simulation introduced above (weak/strong/comprehensive/refinement/etc.), to the corresponding notion *via* (m, n) diagrams (of the relevant kind), rather than via corresponding individual steps. We see examples of all this below.

7 BASIC CONCEDING/RESTORING

We know that for a retrenchment $SysX \succcurlyeq_{[G/W/D]} SysY$, general trace inclusion does not automatically hold. However the notion of (m, n) diagrams of various strengths gives hope that if for $SysX \succcurlyeq_{[G/W/D]} SysY$ a sufficiently copious supply of such diagrams exists, then we can derive a weakened kind of trace inclusion nevertheless.

Definition 7.1 (Restoring Set). *Let $SysX \succcurlyeq_{[G/W/D]} SysY$. Let $UI_Y^R \subseteq \mathsf{U}_Y \times (\biguplus_{Op_Y} \mathsf{I}_{Op_Y})$ be a set of before-states and inputs of $SysY$ such that for any u_Y -(i_Y, Op_Y, o_Y)-> u'_Y where $(u_Y, i_Y) \in UI_Y^R$ and any $u_X \in \mathsf{U}_X$, there is an execution fragment of $SysX$ from u_X to a state $\tilde{u}_X$ such that u_Y -(i_Y, Op_Y, o_Y)-> u'_Y is simulable from $\tilde{u}_X$. Then we say that UI_Y^R is a restoring set for $SysY$.*

Note that although a restoring set exhibits quite a strong property in being able to recover simulability from *any* abstract state, no relationship between the manner of such recovery, and any particular concrete behaviour is promised, so there is no connection with any refinement property via this route.

Theorem 7.2 (Weakly Simulation with Restoring Set). *Let $SysX \succcurlyeq_{[G/W/D]} SysY$. Let UI_Y^R be a restoring set for $SysY$. Let $\mathbf{M}$ be the set of executions YY of $SysY$ such that: for any proper prefix YY^{prpr} of YY, if YY^{prpr} is weakly simulable, but the next step of YY after YY^{prpr} is not simulable from the last abstract state of a weak simulation of YY^{prpr}, then YY visits a state in UI_Y^R in the suffix that follows YY^{prpr}. Then $SysY$ is weakly simulable over $\mathbf{M}$ by $SysX$.*

Proof. Suppose that $u_{Y,0}$ is an initial state of $SysY$ and let $YY \in \mathbf{M}$ be an arbitrary element of $\mathbf{M}$, e.g.:

$$YY \equiv [\, u_{Y,0} \text{-}(i_{Y,0}, Op_{Y,0}, o_{Y,1})\text{->} u_{Y,1} \text{-}(i_{Y,1}, Op_{Y,1}, o_{Y,2})\text{->} u_{Y,2} \ \ldots\,] \tag{32}$$

We build a weak simulation XX of YY by $SysX$. Then, since YY was arbitrary, weak simulation over $\mathbf{M}$ of $SysY$ by $SysX$ follows.

By the initialisation PO (5), there is an initial state of $SysX$, $u_{X,0}$, such that $G(u_{X,0}, u_{Y,0})$ holds. This is the base case of an induction.

For the inductive step, suppose we have weakly simulated the YY up to $u_{Y,k}$, by having built a $SysX$ execution up to a state $u_{X,\tilde{k}}$. There are now three cases.

In the first case, there are no further steps of YY and we are done.

In the second case, there is a next YY step, $u_{Y,k}$ -$(i_{Y,k}, Op_{Y,k}, o_{Y,k+1})$-> $u_{Y,k+1}$, and we have that, for some $Op_{X,\tilde{k}} \succcurlyeq Op_{Y,k}$ and $i_{X,\tilde{k}} \in \mathsf{I}_{Op_{X,\tilde{k}}}$, $W_{Op_{X,Y}}(u_{X,\tilde{k}}, i_{X,\tilde{k}}, u_{Y,k}, i_{Y,k})$ holds. In this case the retrenchment correctness PO (19) states that there is a next abstract step $u_{X,\tilde{k}}$ -$(i_{X,\tilde{k}}, Op_{X,\tilde{k}}, o_{X,\tilde{k}+1})$-> $u_{X,\tilde{k}+1}$, such that the abstract and concrete steps are in simulation, which completes the inductive step for this case.

In the third case, there is a next YY step, $u_{Y,k}$ -$(i_{Y,k}, Op_{Y,k}, o_{Y,k+1})$-> $u_{Y,k+1}$, but it is not simulable from $u_{X,\tilde{k}}$. In this case we know that there is a state $u_{Y,t}$ to the future of $u_{Y,k}$ in YY, that is in UI_Y^R. Then, by the properties of UI_Y^R, we can continue the construction of XX from $u_{X,\tilde{k}}$ by zero or more steps, arriving at a state $u_{X,\tilde{s}}$, such that if there is a step $u_{Y,t}$ -$(i_{Y,t}, Op_{Y,t}, o_{Y,t+1})$-> $u_{Y,t+1}$ of YY starting at $u_{Y,t}$, then there is a state $u_{X,\tilde{t}}$, reachable from $u_{X,\tilde{s}}$, and a step $u_{X,\tilde{t}}$ -$(i_{X,\tilde{t}}, Op_{X,\tilde{t}}, o_{X,\tilde{t}+1})$-> $u_{X,\tilde{t}+1}$ from $u_{X,\tilde{t}}$ which is in simulation with $u_{Y,t}$ -$(i_{Y,t}, Op_{Y,t}, o_{Y,t+1})$-> $u_{Y,t+1}$. We build the weak (m,n)-diagram from $(u_{X,\tilde{k}}, u_{Y,k})$ to the pair $(u_{X,\tilde{s}}, u_{Y,t})$, which completes the inductive step. We are done. □

We regard Theorem 7.2 as a minimal 'recovery from occasional disturbances' theorem. Provided the behaviour of the concrete system is mostly benign, its behaviour remains faithful to the relevant abstraction, by means of the simulation property. However, from time to time, the concrete system may engage in behaviour not allowed for by the abstract system, and then the rather strong assumptions made about UI_Y^R provide the guarantee we need to return to simulation.

Theorem 7.2 works by setting up UI_Y^R as the guarantor of the property that *arbitrary* abstract states can find a way back to simulation. We can imagine many variations on this basic idea by setting up the inevitability of return to simulability in different ways. However, the more complex the construction, the more delicate the justification would need to be that the desired inevitability really

does follow. Essentially, this amounts to an interplay between *angelic* aspects of the construction (embodied in the definition of **M** in Theorem 7.2) and *demonic* aspects (embodied in the universally quantified elements of Theorem 7.2).

One relatively obvious application of this framework is when the concrete system is prone to cope with occasional erroneous or exceptional situations —which are are not modelled at the abstract level— after the handling of which it returns to a known state in order to continue normal processing. The abstract system can be augmented with a generic everywhere enabled *Reset* operation, which does not follow the details of the concrete recovery, but sends the abstract system from the last simulating state into a known abstract state (e.g. an initial state) that can be made to correspond with the concrete one, restoring simulability. If the added *Reset* operation is atomic, then it is invariant preserving, since both the last simulating state and the known abstract state will satisfy the abstract invariant. However, if the *Reset* operation has a duration, more care is needed regarding invariant preservation. Evidently, the full generality of the construction in Theorem 7.2 is not needed in this simple scenario.

8 METRIC CONCEDING/RESTORING

We recall that a metric on a set X is a distance function $d : X \times X \to \mathbb{R}^+ \cup \{\infty\}$, which satisfies the usual laws of identity $d(a,a) = 0$, symmetry $d(a,b) = d(b,a)$, and the triangle inequality $d(a,b) + d(b,c) \geq d(a,c)$.[12] A metric space *with origin* $O \in X$, endows each element $a \in X$ with a magnitude $\mu(a) = d(O,a)$. Normed linear spaces are obvious with-origin examples.

Definition 8.1 (Metric on R-Mediated Union). *For $i \in \{1,2\}$ let X_i carry metric d^i in the sense just stated, and let $R : X_1 \leftrightarrow X_2$ be a relation. We extend d^i on X_i to a metric $d^{1,2}$ on the (disjoint) union $X_1 \uplus X_2$ thus:*

$$d^{1,2}(u_i, \hat{u}_i) \;=\; d^i(u_i, \hat{u}_i) \qquad \text{if} \quad \{u_i, \hat{u}_i\} \subseteq X_i$$

$$d^{1,2}(u_1, u_2) \;=\; \begin{cases} \min\limits_{\tilde{u}_1, \tilde{u}_2} \{\; \frac{1}{2}(d^1(u_1, \tilde{u}_1) + d^2(u_2, \tilde{u}_2)) \;\mid \\ \qquad u_1 \in \mathrm{dom}(R) \wedge R(\tilde{u}_1, u_2) \wedge u_2 \in \mathrm{ran}(R) \wedge R(u_1, \tilde{u}_2) \;\} \\ \infty \qquad\qquad\qquad\qquad \text{otherwise} \end{cases} \tag{33}$$

We call $d^{1,2}$ the R-mediated (disjoint) union of d^1 and d^2 on $X_1 \uplus X_2$. If each X_i has origin O_i each element of $X_1 \uplus X_2$ inherits a magnitude from its underlying component.

We henceforth assume that the state and I/O spaces we deal with are all metric spaces of the kind described, and when they are assumed to have an origin we will always say so explicitly. We apply the construction of Definition 8.1 to relations arising from refinements and retrenchments (and their data), but without writing all the edge cases and side conditions explicitly.

8.1 Metrics, Simulations, Refinements, Onto Retrenchments

For an arbitrary system *SysX*, let the metrics on U_X be d^X, on I_{Op_X} be $d^{I_{Op_X}}$, and on O_{Op_X} be $d^{O_{Op_X}}$. Assume similar notations for other systems.

Definition 8.2 (Standard Metric for System, Refinement, Retrenchment). *Adopting the notational conventions established, and given SysX, SysY and a gluing relation G, we define $d^G_{X,Y}$ thus:*

$$d^G_{X,Y}(u_X, u_Y) \;=\; \min_{\tilde{u}_X, \tilde{u}_Y} \{\; \tfrac{1}{2}(d^X(u_X, \tilde{u}_X) + d^Y(u_Y, \tilde{u}_Y)) \mid G(\tilde{u}_X, u_Y) \wedge G(u_X, \tilde{u}_Y) \;\} \tag{34}$$

[12] We include ∞ as a valid distance between entities that are incompatible in various ways. We also omit the unicity law $d(a,b) = 0 \Rightarrow a = b$, so we are dealing, strictly speaking, with pseudometric spaces. We omit to write 'pseudo' below.

If we now have a refinement SysX $\geqslant_{[G/In/Out]}$ *SysY, we additionally define, for* $Op_X \geqslant Op_Y$*:*

$$d^{In_{Op}}_{X,Y}(i_X, i_Y) \;=\; \min_{\tilde{i}_X, \tilde{i}_Y} \{\; \tfrac{1}{2}(d^{I_{Op_X}}(i_X, \tilde{i}_X) + d^{I_{Op_Y}}(i_Y, \tilde{i}_Y)) \mid In_{Op_{X,Y}}(\tilde{i}_X, i_Y) \wedge In_{Op_{X,Y}}(i_X, \tilde{i}_Y) \;\} \tag{35}$$

$$d^{Out_{Op}}_{X,Y}(o_X, o_Y) \;=\; \min_{\tilde{o}_X, \tilde{o}_Y} \{\; \tfrac{1}{2}(d^{O_{Op_X}}(o_X, \tilde{o}_X) + d^{O_{Op_Y}}(o_Y, \tilde{o}_Y)) \mid Out_{Op_{X,Y}}(\tilde{o}_X, o_Y) \wedge Out_{Op_{X,Y}}(o_X, \tilde{o}_Y) \;\} \tag{36}$$

If, instead, we have a retrenchment SysX $\succcurlyeq_{[G/W/D]}$ *SysY, we analogously define, for* $Op_X \succcurlyeq Op_Y$*:*

$$d^{W_{Op}}_{X,Y}(i_X, i_Y) \;=\; \min_{u_X, u_Y, \tilde{i}_X, \tilde{i}_Y} \{\; \tfrac{1}{2}(d^{I_{Op_X}}(i_X, \tilde{i}_X) + d^{I_{Op_Y}}(i_Y, \tilde{i}_Y)) \mid W_{Op_{X,Y}}(u_X, \tilde{i}_X, u_Y, i_Y) \wedge W_{Op_{X,Y}}(u_X, i_X, u_Y, \tilde{i}_Y) \;\} \tag{37}$$

$$d^{D_{Op}}_{X,Y}(o_X, o_Y) \;=\; \min_{\substack{u_X, u_Y, i_X, i_Y, \\ u'_X, u'_Y, \tilde{o}_X, \tilde{o}_Y}} \{\; \tfrac{1}{2}(d^{O_{Op_X}}(o_X, \tilde{o}_X) + d^{O_{Op_Y}}(o_Y, \tilde{o}_Y)) \mid D_{Op_{X,Y}}(u_X, i_X, u'_X, \tilde{o}_X, u_Y, i_Y, u'_Y, o_Y) \wedge D_{Op_{X,Y}}(u_X, i_X, u'_X, o_X, u_Y, i_Y, u'_Y, \tilde{o}_Y) \;\} \tag{38}$$

The above are also well defined when we merely have refinement or retrenchment data $[G/In/Out]$ *or* $[G/W/D]$*. We call (34)-(38) the standard metrics of the refinement or retrenchment data.*

Definition 8.3 (Standard Associated Metric Refinement, Retrenchment Relations). *Let SysX and SysY be given. Let* $\mathrm{Ops}_X \geqslant \mathrm{Ops}_Y$ *with* $[G/In/Out]$ *constitute refinement data for SysX and SysY. Suppose that the state and I/O spaces are metric, and assume the standard metrics of Definition 8.2. Let:*

$$G^{\Delta_G}_{X,Y}(u_X, u_Y) \;\equiv\; d^{G}_{X,Y}(u_X, u_Y) \leq \Delta_G \tag{39}$$

$$In^{\Delta_I}_{Op_{X,Y}}(i_X, i_Y) \;\equiv\; d^{In_{Op}}_{X,Y}(i_X, i_Y) \leq \Delta_I \tag{40}$$

$$Out^{\Delta_O}_{Op_{X,Y}}(o_X, o_Y) \;\equiv\; d^{Out_{Op}}_{X,Y}(o_X, o_Y) \leq \Delta_O \tag{41}$$

where Δ_G*,* Δ_I *and* Δ_O *are constants. We call these the* standard associated metric refinement relations *for the given refinement data (and given constants). Now, if we assume additionally that all the metric spaces have an origin, we can redefine:*

$$G^{\Delta_G}_{X,Y}(u_X, u_Y) \;\equiv\; \eta\, d^{G}_{X,Y}(u_X, u_Y) + \alpha_X\, \mu(u_X) + \alpha_Y\, \mu(u_Y) \leq \Delta_G \tag{42}$$

$$In^{\Delta_I}_{Op_{X,Y}}(i_X, i_Y) \;\equiv\; \eta\, d^{In_{Op}}_{X,Y}(i_X, i_Y) + \alpha_X\, \mu(i_X) + \alpha_Y\, \mu(i_Y) \leq \Delta_I \tag{43}$$

$$Out^{\Delta_O}_{Op_{X,Y}}(o_X, o_Y) \;\equiv\; \eta\, d^{Out_{Op}}_{X,Y}(o_X, o_Y) + \alpha_X\, \mu(o_X) + \alpha_Y\, \mu(o_Y) \leq \Delta_O \tag{44}$$

In (42)-(44) the constants η*,* α_X*,* α_Y*, are not intended to be global. We have merely suppressed the additional indexing to make them specific to the (pair of) metric spaces in question. Similarly henceforth.*

Alternatively, if $\mathrm{Ops}_X \succcurlyeq \mathrm{Ops}_Y$ *with* $[G/W/D]$ *are retrenchment data for SysX and SysY, we define the* standard associated metric retrenchment relations *for the given retrenchment data (and given constants) as:*

$$G^{\Delta_G}_{X,Y}(u_X, u_Y) \;\equiv\; d^{G}_{X,Y}(u_X, u_Y) \leq \Delta_G \tag{45}$$

$$In^{\Delta_I}_{Op_{X,Y}}(i_X, i_Y) \;\equiv\; d^{W_{Op}}_{X,Y}(i_X, i_Y) \leq \Delta_I \tag{46}$$

$$Out^{\Delta_O}_{Op_{X,Y}}(o_X, o_Y) \;\equiv\; d^{D_{Op}}_{X,Y}(o_X, o_Y) \leq \Delta_O \tag{47}$$

Now, if we assume additionally that all the metric spaces have an origin, we can redefine (with the same proviso as above regarding the constants):

$$G^{\Delta_G}_{X,Y}(u_X, u_Y) \;\equiv\; \eta\, d^{G}_{X,Y}(u_X, u_Y) + \alpha_X\, \mu(u_X) + \alpha_Y\, \mu(u_Y) \leq \Delta_G \tag{48}$$

$$In^{\Delta_I}_{Op_{X,Y}}(i_X, i_Y) \;\equiv\; \eta\, d^{W_{Op}}_{X,Y}(i_X, i_Y) + \alpha_X\, \mu(i_X) + \alpha_Y\, \mu(i_Y) \leq \Delta_I \tag{49}$$

$$Out^{\Delta_O}_{Op_{X,Y}}(o_X, o_Y) \;\equiv\; \eta\, d^{D_{Op}}_{X,Y}(o_X, o_Y) + \alpha_X\, \mu(o_X) + \alpha_Y\, \mu(o_Y) \leq \Delta_O \tag{50}$$

Notation 8.4. *The data defined in (39)-(41), (42)-(44), (45)-(47), (48)-(50), all have the same signature, and in much of the sequel we will rely solely on such data. We will write SysX $\ni$ SysY (with similar notation for the other elements of the relationship between SysX and SysY) to indicate SysX $\geqslant$ SysY and/or SysX $\succcurlyeq$ SysY, when the distinction is not important, and we will say that SysX ref/ret SysY. Similarly, we will write $[G^{\Delta_G}/In^{\Delta_I} \ni Out^{\Delta_O}]$ to refer to the standard metric data derived from some instance of ref/ret data according to Definition 8.2, and will refer to it as the standard metric ref/ret data (arising from the ref/ret data that is to be understood from the discourse).*

If the metric spaces involved have origins, then we redefine the distance functions using $(\eta, \alpha_X, \alpha_Y)$ triples, as above. Thus a reference to a typical $d^{H_{Op}}_{X,Y}(h_X, h_Y)$ distance value becomes a reference to:

$$\eta\, d^{H_{Op}}_{X,Y}(h_X, h_Y) + \alpha_X\, \mu(h_X) + \alpha_Y\, \mu(h_Y) \tag{51}$$

where the $d^{H_{Op}}_{X,Y}$ in (51) is a reference to the original distance function.

In the context of all these variations, we will write $d^{\ni}_{X,Y}$ to denote $d^{In_{Op}}_{X,Y}$ if $\ni$ is $\geqslant$, and to denote $d^{W_{Op}}_{X,Y}$ if $\ni$ is $\succcurlyeq$.

Definition 8.5 (*$EqEnbl_{X,Y}$* Relation). *Let $[G^{\Delta_G}/In^{\Delta_I} \ni Out^{\Delta_O}]$ be the standard metric ref/ret data for SysX and SysY. Define $EqEnbl_{X,Y}$ as follows.*

If $\ni$ is $\geqslant$ then, aggregating over all the operation pairs $Op_X \geqslant Op_Y$:

$$\begin{aligned} &EqEnbl_{X,Y}(u_X, u_Y) \;\equiv \\ &\bigwedge_{Op_X \geqslant Op_Y} \big(In_{Op_{X,Y}}(i_X, i_Y) \Rightarrow \\ &\qquad \big((\exists\, u'_X \in U_X, o_X \in O_X \bullet stp_{Op_X}(u_X, i_X, u'_X, o_X)) \Leftrightarrow \\ &\qquad\quad (\exists\, u'_Y \in U_Y, o_Y \in O_Y \bullet stp_{Op_Y}(u_Y, i_Y, u'_Y, o_Y)) \big) \big) \end{aligned} \tag{52}$$

If $\ni$ is $\succcurlyeq$ then, aggregating over all the operation pairs $Op_X \succcurlyeq Op_Y$:

$$\begin{aligned} &EqEnbl_{X,Y}(u_X, u_Y) \;\equiv \\ &\bigwedge_{Op_X \succcurlyeq Op_Y} \big(W_{Op_{X,Y}}(u_X, i_X, u_Y, i_Y) \Rightarrow \\ &\qquad \big((\exists\, u'_X \in U_X, o_X \in O_X \bullet stp_{Op_X}(u_X, i_X, u'_X, o_X)) \Leftrightarrow \\ &\qquad\quad (\exists\, u'_Y \in U_Y, o_Y \in O_Y \bullet stp_{Op_Y}(u_Y, i_Y, u'_Y, o_Y)) \big) \big) \end{aligned} \tag{53}$$

Theorem 8.6 (Trace Inclusion with Standard Metric Data). *Let $[G^{\Delta_G}/In^{\Delta_I} \ni Out^{\Delta_O}]$ be standard metric ref/ret data constructed from either refinement data or onto retrenchment data for SysX and SysY. Suppose that:*

$$Init_X(u_X) \wedge Init_Y(u_Y) \Rightarrow G^{\Delta_G}_{X,Y}(u_X, u_Y) \wedge EqEnbl_{X,Y}(u_X, u_Y) \tag{54}$$

and, let each operation pair $Op_X \ni Op_Y$ *satisfy:*

$$G^{\Delta_G}_{X,Y}(u_X, u_Y) \wedge In^{\Delta_I}_{Op_{X,Y}}(i_X, i_Y) \wedge stp_{OpX}(u_X, i_X, u'_X, o_X) \wedge stp_{OpY}(u_Y, i_Y, u'_Y, o_Y) \Rightarrow$$
$$(\exists\, \tilde{u}'_X \in \mathsf{U}_X, \tilde{o}_X \in \mathsf{O}_{Op_X} \bullet stp_{OpX}(u_X, i_X, \tilde{u}'_X, \tilde{o}_X) \wedge$$
$$G^{\Delta_G}_{X,Y}(\tilde{u}'_X, u'_Y) \wedge Out^{\Delta_O}_{Op_{X,Y}}(\tilde{o}_X, o_Y) \wedge EqEnbl_{X,Y}(\tilde{u}'_X, u'_Y)) \tag{55}$$

Then there is a trace inclusion property from SysY traces to SysX traces that is mediated by the data $[G^{\Delta_G}/In^{\Delta_I} \ni Out^{\Delta_O}]$.

Furthermore, if all the metric spaces are now assumed to have an origin, then, with suitable reinterpretation of the symbols, the result continues to hold.

Proof. To show the trace inclusion, let $[\ u_{Y,0}\ \text{-}(i_{Y,0}, Op_{Y,0}, o_{Y,1})\!\rightarrow\ u_{Y,1}\ \text{-}(i_{Y,1}, Op_{Y,1}, o_{Y,2})\!\rightarrow\ u_{Y,2}\ \ldots\ ]$ be an execution of *SysY*. We construct a simulating execution of *SysX*. Let $u_{X,0}$ be an initial state of *SysX*. Using (54), we deduce that $G^{\Delta_G}_{X,Y}$ and $EqEnbl_{X,Y}$ both hold for $u_{X,0}$ and $u_{Y,0}$. If there are no further steps of the concrete execution, we are done.

Now suppose we have constructed the simulation up to $u_{Y,k}$ and $u_{X,k}$, for which $G^{\Delta_G}_{X,Y}(u_{X,k}, u_{Y,k})$ and $EqEnbl_{X,Y}(u_{X,k}, u_{Y,k})$ both hold. If there are no further steps of the concrete execution, we are done.

Otherwise, there is a next concrete step $u_{Y,k}\ \text{-}(i_{Y,k}, Op_{Y,k}, o_{Y,k+1})\!\rightarrow\ u_{Y,k+1}$. Since $\ni$ is either refinement or onto retrenchment, relations $\mathrm{Ops}_X \ni \mathrm{Ops}_Y$ and $In_{Op_{k,X,Y}}/W_{Op_{k,X,Y}}$ are both onto. Therefore an Op_X and then an i_X can be found to instantiate the hypotheses needed for (52)/(53). Therefore, since we have the next concrete step $u_{Y,k}\ \text{-}(i_{Y,k}, Op_{Y,k}, o_{Y,k+1})\!\rightarrow\ u_{Y,k+1}$, we deduce the existence of a next abstract step $u_{X,k}\ \text{-}(i_{X,k}, Op_{X,k}, o_{X,k+1})\!\rightarrow\ u_{X,k+1}$. For these two steps, (55) enables us to deduce the existence of $\tilde{u}_{X,k+1}$ and $\tilde{o}_{X,k+1}$ (distinct from $u_{X,k+1}$ and $o_{X,k+1}$ if need be), for which $G^{\Delta_G}(\tilde{u}_{X,k+1}, u_{Y,k+1})$, $Out^{\Delta_O}_{Op_{X,Y}}(\tilde{o}_{X,k+1}, o_{Y,k+1})$ and $EqEnbl_{X,Y}(\tilde{u}_{X,k+1}, u_{Y,k+1})$ all hold, which gives the inductive step for the trace inclusion property required.

Moreover, if all the metric spaces are further assumed to have an origin, then, with suitable reinterpretation of the symbols, the result continues to hold, because the proof just given is insensitive to the internal details of the definition of $G^{\Delta_G}_{X,Y}$, $In^{\Delta_I}_{Op_{X,Y}}$, $Out^{\Delta_O}_{Op_{X,Y}}$. □

The above result allows us to introduce some inaccuracy into the behaviour of a system, provided this does not spread over time. There are a couple of further things worth noting.

A simple one is that we make little use of the outputs. These are assumed to be emitted to the environment, and play no further part in the proceedings. The point of view of this paper is that outputs are write-only, and thus no future property *of the model under consideration* can depend on them, so they need not concern us unduly. Of course, in a wider context, outputs are typically sensed by *some other system* and thus their properties are of interest. But in that case, they are part of the state of a combined system including both the original model and the other system, and in such an eventuality, they may be included in the state-centric reasoning of that combined system, and thereby fall under the scope of our work. The issue thus reduces to a matter of nomenclature. We maintain the same point of view in the rest of the paper.

Another is that we do *not* assert either a refinement $SysX \geqslant SysY$ or an onto retrenchment $SysX \succcurlyeq SysY$. For either to hold, it would have to be the case that for *every* $G^{\Delta_G}_{X,Y}$-related pair (u_X, u_Y), enabledness at u_Y would have to imply enabledness at u_X. But in many typical applications, the metric spaces in which to naturally embed an application model are topologically complete, and when one or both of the systems in question is/are discretized, transitions will not issue from *all* states satisfying a simple metric bound, even if there are many states satisfying the bound from

which they *do* do so. So we have to be more circumspect if we want our outcome to indicate the character of more general results to follow.

Yet another is that the theorem covers both the case that there is 'only a little' nondeterminism in the step relations of *SysX* and *SysY* (to allow, e.g., for some inaccuracy arising from discretization, and allowing the identification of $\tilde{u}'_X$ and $\tilde{o}_X$ with u'_X and o_X respectively), and also the case that there is 'more than a little' nondeterminism in these step relations (where $\tilde{u}'_X$ and $\tilde{o}_X$ distinct from u'_X and o_X would be required).

Definition 8.7 (Metric Neighbourhood). *Let SysX, equipped with metrics as in Definition 8.2, be given. Let SysY be an isomorphic copy of SysX and let us write* $\mathbb{i} : SysX \to SysY$ *for all elements of the bijection between SysX and SysY, including state and I/O spaces, operations sets and names, etc. Let constants* Δ_G, Δ_I *and* Δ_O *be given. For each* $u_X \in Rch\mathsf{U}_X$, *let* $nh(\mathbb{i}(u_X)) \subseteq \mathsf{U}_Y$ *be a set of states satisfying:*

$$\mathbb{i}(u_X) \in nh(\mathbb{i}(u_X)) \tag{56}$$

$$u_Y \in nh(\mathbb{i}(u_X)) \Rightarrow d^Y(u_Y, \mathbb{i}(u_X)) < \Delta_G \tag{57}$$

where d^X *is the distance function on* U_X *and* d^Y *is the distance function on* U_Y *inherited from* d^X *via* $\mathbb{i}$. *Make analogous definitions of* $nh(\mathbb{i}(i_X)) \subseteq \mathsf{I}_Y$ *using* $d^{I_{Op}}$ *and* Δ_I *for* $i_X \in In_{Op_X}$, *and of* $nh(\mathbb{i}(o_X)) \subseteq \mathsf{O}_Y$ *using* $d^{O_{Op}}$ *and* Δ_O *for* $o_X \in Out_{Op_X}$. *Let the initial states of SysY be the all states in* $nh(\mathbb{i}(u_X))$ *where* u_X *is an initial state of SysX. Enhance the transition relation of SysY by including all steps* stp_{Op_Y} *satisfying:*

$$\begin{aligned} &Op_Y = \mathbb{i}(Op_X) \wedge u_Y \in nh(\mathbb{i}(u_X)) \wedge i_Y \in nh(\mathbb{i}(o_X)) \wedge u'_Y \in nh(\mathbb{i}(u'_X)) \wedge o_Y \in nh(\mathbb{i}(o_X)) \Rightarrow \\ &\quad (stp_{Op_X}(u_X, i_X, u'_X, o_X) \Leftrightarrow stp_{Op_Y}(u_Y, i_Y, u'_Y, o_Y)) \end{aligned} \tag{58}$$

It is now easy to show:

Theorem 8.8 (Trace Inclusion with Metric Neighbourhood). *Let SysX and SysY be as in Definition 8.7. With the standard metric refinement relations for the data relating SysX and SysY, the conditions of Theorem 8.6 are satisfied, and so there is a trace inclusion property from SysY traces to SysX traces that is mediated by the given data.*

Theorem 8.8 expresses a very abstract kind of discretization process, in which the ideal system behaviour of *SysX* can be approximated by transitions between the states in the neighbourhoods $nh(\mathbb{i}(u_X))$ in *SysY* of reachable states u_X of the ideal behaviour. In this result, all details of the discretization process have been abstracted away.

Equally easy is:

Theorem 8.9 (Refinement with Metric Neighbourhood Refinement Data). *Let SysX and SysY be as in Definition 8.7. Let:*

$$G^\circ_{X,Y}(u_X, u_Y) \equiv u_Y \in nh(\mathbb{i}(u_X)) \tag{59}$$

$$In^\circ_{Op_{X,Y}}(i_X, i_Y) \equiv i_Y \in nh(\mathbb{i}(i_X)) \tag{60}$$

$$Out^\circ_{Op_{X,Y}}(o_X, o_Y) \equiv o_Y \in nh(\mathbb{i}(o_X)) \tag{61}$$

Then there is a refinement from SysX to SysY mediated by refinement data $[G^\circ/In^\circ/Out^\circ]$.

Theorem 8.9 follows because the data $[G^\circ/In^\circ/Out^\circ]$ are restricted to exactly the sources and targets of transitions in the two systems. This allows the requirements of the quantifications in the refinement correctness PO (6) to be satisfied. Thus (*c.f.* above), whenever the hypotheses of the refinement correctness PO are true, a suitable abstract step can always be found.

8.2 Contractions, Simulations, Refinements, Onto Retrenchments

Let X be a metric space and let $T : X \leftrightarrow X$ be a relation on X. Then T is *contracting* iff for all $\{x, y, x', y'\} \subseteq T$, where $T(x, x')$ and $T(y, y')$ both hold, there is a $\kappa < 1$ such that $d(x', y') \leq \kappa\, d(x, y)$. If we can interchange the existential and universal quantification in the preceding, we say that T is uniformly contracting. If X is metrically complete,[13] then the well known fixpoint theorem says that there is a unique fixpoint $\mathsf{FP} \in X$ to which iterated application of a uniformly contracting and total T to any $x \in X$ tends. We apply these ideas to refinements and retrenchments and (their data) in which operations are mostly well behaved, but which at times may behave in an undesirable manner. In such abnormal episodes, the contracting property helps return system behaviour to the desired norms if they have been breached.

Often, rather than having contraction maps that are directly usable in the sense we need, we have relations in the individual systems of a refinement or retrenchment (or data) which need to be combined. The next definition addresses this.

Definition 8.10 (*R*-Adapted Union of Contracting Transition Relations). *Assume the conventions of Definition 8.1 concerning X_i, d^i, R, $d^{1,2}$. Let T^i be contracting on X_i in the sense just given. We extend T^i on X_i to a relation $T^1 \oplus T^2$ on the (disjoint) union $X_1 \uplus X_2$ thus:*

$$T^1 \oplus T^2(u_i, u_i') \;\equiv\; T^i(u_i, u_i') \qquad \text{provided} \quad \{u_i, u_i'\} \subseteq X_i \tag{62}$$

and we refer to $T^1 \oplus T^2$ as the (disjoint) union of T^1 and T^2. Suppose there is a $\kappa < 1$ such that:

$$T^1(u_1, u_1') \wedge T^2(u_2, u_2') \wedge R(\tilde{u}_1, u_2) \Rightarrow \exists \tilde{u}_1' \bullet R(\tilde{u}_1', u_2') \wedge d^1(u_1', \tilde{u}_1') \leq \kappa\, d^1(u_1, \tilde{u}_1) \tag{63}$$

$$T^1(u_1, u_1') \wedge T^2(u_2, u_2') \wedge R(u_1, \tilde{u}_2) \Rightarrow \exists \tilde{u}_2' \bullet R(u_1', \tilde{u}_2') \wedge d^1(u_2', \tilde{u}_2') \leq \kappa\, d^2(u_2, \tilde{u}_2) \tag{64}$$

We call a $T^1 \oplus T^2$ satisfying (63)-(64) an R-adapted union of T^1 and T^2.

Proposition 8.11 (*R*-Adapted Union of Contracting Transition Relations is Contracting). *Let $T^1 \oplus T^2$ be an R-adapted union of T^1 and T^2 as in Definition 8.10. Then $T^1 \oplus T^2$ is a contraction with respect to the metric $d^{1,2}$.*

Proof. For $i \in \{1, 2\}$, for the case that $\{u_i, \hat{u}_i, u_i', \hat{u}_i'\} \subseteq X_i$, with $T^i(u_i, u_i')$ and $T^i(\hat{u}_i, \hat{u}_i')$ both holding, then $d^{1,2}(u_i', \hat{u}_i') = d^i(u_i', \hat{u}_i') \leq \kappa\, d^i(u_i, \hat{u}_i) = d^{1,2}(u_i, \hat{u}_i)$ for some $\kappa < 1$. This is sufficient for that case.

For the case that $\{u_1, u_1'\} \subseteq X_1$, $\{u_2, u_2'\} \subseteq X_2$, $d^{1,2}(u_1, u_2) < \infty$, and $T^1(u_1, u_1')$, $T^2(u_2, u_2')$:

$$\begin{aligned} d^{1,2}(u_1', u_2') &= \min_{\tilde{u}_1', \tilde{u}_2'} \{\, \tfrac{1}{2}(d^1(u_1', \tilde{u}_1') + d^2(u_2', \tilde{u}_2')) \mid R(\tilde{u}_1', u_2') \wedge R(u_1', \tilde{u}_2') \,\} \\ &< \kappa \min_{\tilde{u}_1, \tilde{u}_2} \{\, \tfrac{1}{2}(d^1(u_1, \tilde{u}_1) + d^2(u_2, \tilde{u}_2)) \mid R(\tilde{u}_1, u_2) \wedge R(u_1, \tilde{u}_2) \,\} \\ &\qquad \text{for some } \kappa < 1, \text{ because of (63)-(64)} \\ &= d^{1,2}(u_1, u_2) \end{aligned} \tag{65}$$

This is sufficient to complete the proof. □

The conditions (63)-(64), though sufficient, are quite demanding. For example, for uniformly contracting T^i, if the fixpoints $\mathsf{FP}_i \in X_i$ do not satisfy $R(\mathsf{FP}_1, \mathsf{FP}_2)$, then (63)-(64) are bound to fail somewhere.

[13] We intend completeness in the sense that all Cauchy sequences in X reach a limit in X.

Theorem 8.12 (Trace Inclusion with Contracting Metric Data). *Let $\kappa < (1 + \Delta_I/\Delta_G)^{-1}$ be a constant. Assume the hypotheses of Theorem 8.6, but, for each operation pair $Op_X \ni Op_Y$ replacing (55) by:*

$$\begin{aligned} & G^{\Delta_G}_{X,Y}(u_X, u_Y) \wedge In^{\Delta_I}_{Op_{X,Y}}(i_X, i_Y) \wedge stp_{Op_X}(u_X, i_X, u'_X, o_X) \wedge stp_{Op_Y}(u_Y, i_Y, u'_Y, o_Y) \Rightarrow \\ & \quad (\exists\, \tilde{u}'_X \in \mathsf{U}_X, \tilde{o}_X \in \mathsf{O}_{Op_X} \bullet stp_{Op_X}(u_X, i_X, \tilde{u}'_X, \tilde{o}_X) \wedge \\ & \qquad d^G_{X,Y}(\tilde{u}'_X, u'_Y) \leq \kappa\, (d^G_{X,Y}(u_X, u_Y) + d^{\ni}_{X,Y}(i_X, i_Y)) \wedge \\ & \qquad Out^{\Delta_O}_{Op_{X,Y}}(\tilde{o}_X, o_Y) \wedge EqEnbl_{X,Y}(\tilde{u}'_X, u'_Y)) \end{aligned} \tag{66}$$

Then there is a trace inclusion property from SysY traces to SysX traces that is mediated by the data $[G^{\Delta_G}/In^{\Delta_I} \ni Out^{\Delta_O}]$. The result holds if all the metric spaces involved do not have origins, or if they do, provided the relevant distance and relation symbols are interpreted correctly.

Proof. The only part of the proof of Theorem 8.6 that needs attention involves a detail of the inductive step. For the concrete and abstract transitions identified, $u_{Y,k}$ -$(i_{Y,k}, Op_{Y,k}, o_{Y,k+1})$-> $u_{Y,k+1}$ and $u_{X,k}$ -$(i_{X,k}, Op_{X,k}, o_{X,k+1})$-> $u_{X,k+1}$, assuming (66) enables us to infer that, since $\kappa < (1+\Delta_I/\Delta_G)^{-1}$, assuming $G^{\Delta_G}_{X,Y}(u_X, u_Y)$ implies that $G^{\Delta_G}_{X,Y}(\tilde{u}_X, u'_Y)$ will hold. The remainder of the proof is the same. □

Theorem 8.12 allows for the fact that the presence of an input will affect the execution of any transition, potentially increasing the 'effective distance' between concrete and abstract before-states/inputs. The assumed $\kappa < (1+\Delta_I/\Delta_G)^{-1}$ allows for this to the extent that the effective distance is assumed to be $d^G_{X,Y}(u_X, u_Y) + d^{\ni}_{X,Y}(i_X, i_Y)$ which is bounded by $\Delta_G + \Delta_I$. The result is modular among the no-origin/with-origin options.

It is now not hard to imagine that with increasingly complex assumptions about the metrics, I/O and state behaviour, one could deduce increasingly complex results of a similar kind.

In the same vein, for the constructions based on metric neighbourhoods we find:

Theorem 8.13 (Trace Inclusion with Contracting Metric Neighbourhood). *Incorporating the modifications of Theorem 8.12 into the assumptions of Theorem 8.8 (Trace Inclusion with Metric Neighbourhood), the theorem continues to hold, namely that with the standard metric refinement relations for the data relating SysX and SysY, there is a trace inclusion property from SysY traces to SysX traces that is mediated by the given data.*

Theorem 8.14 (Refinement with Contracting Metric Neighbourhood Refinement Data). *Incorporating the modifications of Theorem 8.12 into the assumptions of Theorem 8.9 (Refinement with Metric Neighbourhood Refinement Data), the theorem continues to hold, namely that there is a refinement from SysX to SysY mediated by refinement data $[G^\circ/In^\circ/Out^\circ]$.*

8.3 Contractions, Simulations, Retrenchments

The approach of the previous section worked because we assumed absolute bounds on distances in the state and I/O spaces. In some situations this is justifiable, in others not. We now consider situations in which there are some regions in which the assumption is valid as well as other regions in which it is not.

Definition 8.15 (Diverging/Converging Framework). *Let $[G^{\Delta_G}/In^{\Delta_I} \ni Out^{\Delta_O}]$ be standard metric ref/ret data constructed from either refinement data or onto retrenchment data for SysX and SysY.*

Let $\mathbb{R}^+_{\neg 1} \equiv \mathbb{R}^+ - \{1\}$ be the non-negative reals without 1.

Let $\mathcal{K} \subseteq \mathbb{R}^+_{\neg 1}$ be a finite set of positive constants. For $k \in \mathcal{K}$, let $B_k \in \mathbb{R}^+$ be a positive constant, and let $UI^k_{XY} \subseteq (\mathsf{U}_X \times \mathsf{U}_Y) \times \biguplus_{Op_X \geqslant Op_Y} (\mathsf{I}_{Op_X} \times \mathsf{I}_{Op_Y})$ be a set of tuples of states and inputs of SysX and SysY such that:

$$
\begin{array}{l}
(u_X, u_Y, i_X, i_Y) \in UI^k_{XY} \Rightarrow \\
\quad stp_{Op_X}(u_X, i_X, u'_X, o_X) \wedge stp_{Op_Y}(u_Y, i_Y, u'_Y, o_Y) \Rightarrow \\
\quad d^{\ni}_{X,Y}(i_X, i_Y) \leq B_k \wedge \\
\quad\quad (\exists\, \tilde{u}'_X \in \mathsf{U}_X, \tilde{o}_X \in \mathsf{O}_{Op_X} \bullet stp_{Op_X}(u_X, i_X, \tilde{u}'_X, \tilde{o}_X) \wedge \\
\quad\quad\quad d^G_{X,Y}(\tilde{u}'_X, u'_Y) \leq k(d^G_{X,Y}(u_X, u_Y) + d^{\ni}_{X,Y}(i_X, i_Y)) \wedge \\
\quad\quad\quad EqEn_{X,Y}(\tilde{u}'_X, u'_Y))
\end{array} \tag{67}
$$

where

$$
EqEn_{X,Y}(u_X, u_Y) \;\equiv\; \bigvee_{k \in \mathcal{K}} EqEn^k_{X,Y}(u_X, u_Y) \tag{68}
$$

and where

$$
\begin{array}{l}
EqEn^k_{X,Y}(u_X, u_Y) \;\equiv \\
\quad [\; (\exists\, i_X \in \mathsf{I}_X, u'_X \in \mathsf{U}_X, o_X \in \mathsf{O}_X \bullet stp_{Op_X}(u_X, i_X, u'_X, o_X)) \Leftrightarrow \\
\quad\quad (\exists\, i_Y \in \mathsf{I}_Y, u'_Y \in \mathsf{U}_Y, o_Y \in \mathsf{O}_Y \bullet stp_{Op_Y}(u_Y, i_Y, u'_Y, o_Y))\;] \wedge \\
\quad [\; stp_{Op_X}(u_X, i_X, u'_X, o_X) \wedge stp_{Op_Y}(u_Y, i_Y, u'_Y, o_Y) \Rightarrow (u_X, u_Y, i_X, i_Y) \in UI^k_{XY}\;]
\end{array} \tag{69}
$$

Let:

$$
\begin{array}{l}
stp^k_Y \;\equiv\; \{\; stp_{Op_Y}(u_Y, i_Y, u'_Y, o_Y) \mid (\exists\, u_X \bullet EqEn^k_{X,Y}(u_X, u_Y)) \wedge \\
\qquad\qquad\qquad\qquad \overline{k} < k \Rightarrow \neg(\exists\, u_X \bullet EqEn^{\overline{k}}_{X,Y}(u_X, u_Y))\; \}
\end{array} \tag{70}
$$

Suppose also that:

$$
Init_X(u_X) \wedge Init_Y(u_Y) \Rightarrow G^{\Delta_G}_{X,Y}(u_X, u_Y) \wedge EqEn_{X,Y}(u_X, u_Y) \tag{71}
$$

where $G^{\Delta_G}_{X,Y}$ is given by (39) with Δ_G a constant. This collection of properties is called a diverging/converging framework for SysX and SysY, where the stp^k_Y components with $1 < k \in \mathcal{K}$ characterise the diverging aspect, and the stp^k_Y components with $1 > k \in \mathcal{K}$ characterise the converging aspect.

Note that (67) and (68) are mutually recursive, necessitating the calculation of fixpoints in the general case. We remark on this issue again in the context of our main example.

Theorem 8.16 (Trace Inclusions with Diverging/Converging Framework). *Suppose given a diverging/converging framework for SysX and SysY, assuming the notations of Definition 8.15, and where $\mathcal{K} = \{\kappa, K\}$, with $0 < \kappa < 1 < K$. Let $\mathbf{M}$ be a set of executions of SysY that satisfy the* (M, n)-bounded excursion property *(where M and n are both constants), namely that, for every execution $\mathbf{YY}$ in $\mathbf{M}$:*

(i) *All steps are in $stp^\kappa_Y \cup stp^K_Y$.*
(ii) *Every stp^K_Y step of $\mathbf{YY}$ is a member of a subsequence of at most M consecutive such steps.*
(iii) *Every stp^κ_Y step of $\mathbf{YY}$ is a member of a subsequence of at least n consecutive such steps.*

Let $In^B_{Op_{X,Y}}$ and $Out^B_{Op_{X,Y}}$ be given by (72)-(74), and let $G^B_{X,Y}$ be given by (84), all below. Then:

(a) *If each trace in $\mathbf{M}$ starts with at least one stp^K_Y step, and M and n are related by (85) below, then there is an Init-constrained trace inclusion from the traces in $\mathbf{M}$ to traces of SysX, mediated by data $[G^B / In^B / Out^B]$ and Init-constraint $G^{\Delta_G}_{X,Y}$.*

(b) *If each trace in* **M** *starts with at least one* stp^{κ}_{Y} *step, and* M *and* n *are related by (86) below, then there is a trace inclusion from the traces in* **M** *to traces of SysX, mediated by data* $[G^{\Delta_G}/In^{B}/Out^{B}]$.

The result holds if all the metric spaces involved do not have origins, or if they do, provided the relevant distance and relation symbols are interpreted correctly.

Proof. Let $\boldsymbol{YY}$ be an execution of $SysY$. We first construct $\boldsymbol{XX}$, a generic execution of $SysX$ that partly satisfies the conditions for simulating $\boldsymbol{YY}$, and then address the missing conditions under various scenarios, in particular, estimating the variation in $d^{G}_{X,Y}$.

Thus, let $u_{X,0}$ and $u_{Y,0}$ be initial states. By (71), $EqEnbl_{X,Y}$ holds for them. If they are both disabled, we are done with constructing $\boldsymbol{XX}$ and the partial simulation. Otherwise, both are enabled and there are steps issuing from both $u_{X,0}$ and $u_{Y,0}$. The one issuing from $u_{Y,0}$ is the next step of $\boldsymbol{YY}$. From (68) and (69) we know that we can choose the next $\boldsymbol{XX}$ step so that $(u_{X,0}, u_{Y,0}, i_{X,0}, i_{Y,0}) \in UI^{k}_{XY}$ for the relevant k. Now (67) ensures $EqEn_{X,Y}$ for the after-states of these steps, and that gives us the inductive step for progressing the construction of $\boldsymbol{XX}$. The general inductive step follows the preceding argument, but simply starts from the latest pair of after-states constructed.

To be able to establish the claims we need the relevant ref/ret data. Let u_Y -(i_Y, Op_Y, o_Y)-> u'_Y and u_X -(i_X, Op_X, o_X)-> u'_X be a pair of corresponding steps of $\boldsymbol{YY}$ and $\boldsymbol{XX}$. If the former is in stp^{K}_{Y}, we define $In^{B}_{X,Y}$ for the operations $Op_X \ni Op_Y$ via:

$$In^{B}_{Op_{X,Y}}(i_X, i_Y) \;\equiv\; d^{\ni}_{X,Y}(i_X, i_Y) \leq B_K \tag{72}$$

and, for any $Op_X \ni Op_Y$ not already covered by (72), we define $In^{B}_{X,Y}$ via:

$$In^{B}_{Op_{X,Y}}(i_X, i_Y) \;\equiv\; d^{\ni}_{X,Y}(i_X, i_Y) \leq B_\kappa \tag{73}$$

In both cases, following the policy stated earlier, we can define $Out^{B}_{X,Y}$ via:

$$Out^{B}_{Op_{X,Y}}(i_X, i_Y) \;\equiv\; \mathsf{true} \tag{74}$$

It remains for $d^{G}_{X,Y}$ to be estimated. Consider a starting value d_0 of $d^{G}_{X,Y}$, and that stp^{K}_{Y} steps are then executed. After one step, $d^{G}_{X,Y}$ is bounded by $K(d_0 + B_K)$. After two steps, $d^{G}_{X,Y}$ is bounded by $K(K(d_0 + B_K) + B_K)$, and after M steps, since $K > 1$, we have:

$$d^{G}_{X,Y} \leq B_{\mathrm{M}}(d_0, M) = d_0\, K^{M} + B_K(K^{M} + K^{M-1} + \ldots + K) \leq K^{M}(d_0 + MB_K) \tag{75}$$

Thus:

$$B_{\mathrm{M}}(d_0, M) \leq K^{M}(d_0 + MB_K) \tag{76}$$

Likewise if, starting from d_0, stp^{κ}_{Y} steps are executed, since $\kappa < 1$, after n steps, we have:

$$\begin{aligned} d^{G}_{X,Y} \leq B_{\mathrm{n}}(d_0, n) \;&=\; d_0\, \kappa^{n} + B_\kappa(\kappa + \kappa^{2} + \ldots + \kappa^{n}) \\ &=\; d_0\, \kappa_n + B_\kappa(1 - \kappa^{n})/(1/\kappa - 1)^{-1} \\ &\leq\; d_0\, \kappa^{n} + B_\kappa\kappa(1 - \kappa)^{-1} \end{aligned} \tag{77}$$

so that:

$$B_{\mathrm{n}}(d_0, n) \leq d_0\, \kappa^{n} + B_\kappa\kappa(1 - \kappa)^{-1} \tag{78}$$

Note that if $B_{\mathrm{n}}(d_0, n)$ is required to be smaller than d_0, then we must have:

$$d_0 - B_\kappa\kappa(1 - \kappa)^{-1} > d_0\, \kappa^{n} > 0 \tag{79}$$

and if the value of n needed to achieve $B_{\mathrm{n}}(d_0, n) < d_0$ is required to be reasonable, then the difference in (79) should be substantial.

Combining the preceding, if, starting from d_0, M stp_Y^K steps are followed by n stp_Y^κ steps, the final value of $d_{X,Y}^G$ is bounded by:

$$B_{\mathrm{M,n}}(d_0, M, n) \leq K^M(d_0 + MB_K)\kappa^n + B_\kappa\kappa(1-\kappa)^{-1} \tag{80}$$

Note that (80) implies that $B_\kappa\kappa(1-\kappa)^{-1}$ is a useful minimal assumption for the bound for $B_{\mathrm{M,n}}(\_, \_, n)$. Thus, to guarantee to reduce $B_{\mathrm{M,n}}(d_0, M, n)$ to a given value $d_1 > B_\kappa\kappa(1-\kappa)^{-1}$ (based on our assumptions and estimates), we must have $K^M(d_0 + MB_K)\kappa^n + B_\kappa\kappa(1-\kappa)^{-1} < d_1$. This implies that we need a number of stp_Y^κ steps n given by:

$$n \geq \log_\kappa \left[\frac{d_1 - B_\kappa\kappa(1-\kappa)^{-1}}{K^M(d_0 + MB_K)} \right] \tag{81}$$

which is suitably well defined only if the numerator in (81) is positive.

Conversely, if, starting from d_0, n stp_Y^κ steps are followed by M stp_Y^K steps, the final value of $d_{X,Y}^G$ is bounded by:

$$B_{\mathrm{n,M}}(d_0, n, M) \leq K^M(d_0\,\kappa^n + B_\kappa\kappa(1-\kappa)^{-1} + MB_K) \tag{82}$$

Note that (82) implies that $d_0\,\kappa^n + B_\kappa\kappa(1-\kappa)^{-1} + B_K$ is a useful minimal assumption for the bound for $B_{\mathrm{n,M}}(\_, \_, M)$ when at least one stp_Y^K step is contemplated. Thus, to guarantee that $B_{\mathrm{n,M}}(d_0, n, M)$ does not exceed a given value $d_1 > d_0\,\kappa^n + B_\kappa\kappa(1-\kappa)^{-1} + B_K$ (based on our assumptions and estimates), we must have $K^M(d_0\,\kappa^n + B_\kappa\kappa(1-\kappa)^{-1} + MB_K) < d_1$. This implies that we need a number of stp_Y^K steps M which is no more than what is permitted by:

$$1 \leq M \leq \log_K[d_1/(d_0 + B_\kappa\kappa(1-\kappa)^{-1} + MB_K)] \tag{83}$$

This is well defined only if the argument of the logarithm in (83) is big enough — if M is to be significant, then d_1 needs to be large.

Define:

$$G_{X,Y}^B(u_X, u_Y) \;\equiv\; d_{X,Y}^G(u_X, u_Y) \leq K^M(\Delta_G + MB_K) \tag{84}$$

We now consider a number of scenarios.

In a first scenario, suppose that $\boldsymbol{YY}$ begins with stp_Y^K steps and that inputs and outputs are constrained as stated above. Then the maximum value of $d_{X,Y}^G(u_X, u_Y)$ that can be achieved arises when the initial value is at its greatest, namely Δ_G, and there are also at most the maximum number of stp_Y^K steps, namely M of them. By (76), this is bounded by $B_{\mathrm{M}}(\Delta_G, M) = K^M(\Delta_G + MB_K)$, which is $G_{X,Y}^B$. If the number, n, of stp_Y^κ steps that follows, satisfies:

$$n > \log_\kappa \left[\frac{\Delta_G - B_\kappa\kappa(1-\kappa)^{-1}}{K^M(\Delta_G + MB_K)} \right] \tag{85}$$

then $d_{X,Y}^G$ is reduced to at most its initial value, Δ_G. This is the worst case for a round of stp_Y^K steps followed by stp_Y^κ steps. Subsequent rounds can do no worse, and thus the requirements of an *Init*-constrained trace inclusion from the traces in $\mathbf{M}$ to traces of *SysX*, mediated by data $[G^B/In^B/Out^B]$ and *Init*-constraint $G_{X,Y}^{\Delta_G}$ are established. This substantiates the claim in (a).

In a second scenario, suppose that $\boldsymbol{YY}$ begins with stp_Y^κ steps and that inputs and outputs are constrained as stated above. Then the maximum value of $d_{X,Y}^G(u_X, u_Y)$ that can be achieved arises when the initial value is at its greatest, namely Δ_G, and when there are at least the minimum number of stp_Y^κ steps, namely n of them. By (77), this is bounded by $B_{\mathrm{n}}(\Delta_G, n) = \Delta_G\,\kappa^n + B_\kappa\kappa(1-\kappa)^{-1}$. If the number, M, of stp_Y^K steps that follows, satisfies:

$$1 \leq M \leq \log_K[\Delta_G/(\Delta_G + B_\kappa\kappa(1-\kappa)^{-1} + MB_K)] \tag{86}$$

then $d^G_{X,Y}$ increases to at most its initial value, Δ_G. This is the worst case for a round of stp^κ_Y steps followed by stp^K_Y steps. Subsequent rounds can do no worse, and thus the requirements of a trace inclusion from the traces in **M** to traces of *SysX*, mediated by data $[G^{\Delta_G}/In^B/Out^B]$ are established. This substantiates the claim in (b).

As before, the indifference to the internal details of distance functions used above, means the result is equally valid for no-origin and with-origin versions. We are done. □

Definition 8.17 (Unbounded Diverging/Converging Framework). *An unbounded diverging/converging framework for SysX and SysY is defined as in Definition 8.15, but with the removal of the conjunct $(d^{\ni}_{X,Y}(i_X, i_Y) \leq B_k)$ from (67), with the resulting term denoted $UI^{\circ k}_{XY}$. All terms derived directly or indirectly from $UI^{\circ k}_{XY}$ also acquire a $^\circ$ superscript thus: $EqEn_{X,Y} \rightarrow EqEn^\circ_{X,Y}$, $EqEn^k_{X,Y} \rightarrow EqEn^{\circ k}_{X,Y}$, $stp^k_Y \rightarrow stp^{\circ k}_Y$, etc.*

Theorem 8.18 (Simulation, Refinement, Retrenchment with (Unbounded) Diverging/Converging Framework). *Let $\mathcal{K} = \{\kappa, K\}$, with $0 < \kappa < 1 < K$. Let SysX and SysY be given, and let the following hold: $\mathrm{Ops}_X = \{Op\kappa_X, OpK_X\}$ and $\mathrm{Ops}_Y = \{Op\kappa_Y, OpK_Y\}$. Assume the following retrenchment data: $\succcurlyeq_{Ops_{X,Y}} = \{Op\kappa_X \succcurlyeq Op\kappa_Y, OpK_X \succcurlyeq OpK_Y\}$, and for $k \in \mathcal{K}$ let:*

$$W^k_{Opk_{X,Y}} \equiv \mathrm{true} \tag{87}$$

$$D^k_{Opk_{X,Y}} \equiv d^G_{X,Y}(u'_X, u'_Y) \leq k(d^G_{X,Y}(u_X, u_Y) + d^{W_{Op}}_{X,Y}(i_X, i_Y)) \tag{88}$$

Suppose given an unbounded diverging/converging framework for SysX and SysY, assuming the notations of Definitions 8.15 and 8.17. Assume (71) (i.e. $G^{\Delta_G}_{X,Y} \wedge EqEn_{X,Y}$ for paired initial states). Then:

(a) *SysY is strongly comprehensively simulable by SysX, mediated by data $[G^{\Delta_G}/W^k_{Opk}/D^k_{Opk}]$.*

(b) *If the conjunct $(d^{In_{Op}}_{X,Y}(i_X, i_Y) \leq B_k)$ is reinstated in the definition of $UI^{\circ k}_{XY}$ (so that the diverging/converging framework is no longer unbounded), and the properties of M, n and G^B are assumed as in Theorem 8.16, then SysY is Init-constrained refinement simulable by SysX, mediated by data $[G^B/W^k_{Opk}/D^k_{Opk}]$ and Init-constraint G^{Δ_G}.*

(c) *Taking all possible SysX simulations **XX** of all possible SysY traces **YY** into account (as constructed for parts (a) and (b)), if $[G^B/W^k_{Opk}/D^k_{Opk}]$ are restricted to just those values that occur within these simulations, then there is a retrenchment from SysX to SysY with these restricted data.*

The result holds if all the metric spaces involved do not have origins, or if they do, provided the relevant distance and relation symbols are interpreted correctly.

Proof. Since we have a(n unbounded) diverging/converging framework for *SysX* and *SysY*, we can rerun the initial part of the proof of Theorem 8.16, to construct a simulating trace **XX** for each *SysY* trace **YY**, since the conjunct $(d^{\ni}_{X,Y}(i_X, i_Y) \leq B_k)$, present in the definition of UI^k_{XY} but absent in the definition of $UI^{\circ k}_{XY}$, is not needed for that. The stated claims now follow by interpreting this fact in the following manner.

For (a), we observe that for each corresponding pair of steps in simulating **XX** and **YY**, $W^k_{Opk_{X,Y}}$ and $D^k_{Opk_{X,Y}}$ are in fact true, and that is all we need.

For (b), we observe that with the reinstated facts, the argument about the maximum value of $d^G_{X,Y}$ detailed in the proof of Theorem 8.16 can be rerun, and this is enough to establish the claim.

For (c), we merely note that whenever we are required, by the retrenchment correctness PO, to prove some consequence on the basis of some hypotheses, that consequence has already been established by the simulation from which the truth of the hypotheses has itself been established

during the restriction process. The independence of the argument from the details of the distance functions implies equal validity of no-origin and with-origin versions of the result. We are done. □

8.4 Allowing for Limited Precision and Noise

The preceding results worked to 'unlimited precision' in that no limit was placed on the precision of the contractions that govern how the two systems converge in the $\kappa < 1$ case, which is a little unrealistic. We now place some boundaries on this behaviour by introducing a threshold below which further contraction is not asserted. This approach not only caters for the limited precision inherent in all real systems, but also yields a means to make some allowance for the equipment and environmental noise that invariably affects all real systems to a greater or lesser extent.

Definition 8.19 (Threshold Diverging/ Converging Framework). *Let* $[G^{\Delta_G}/In^{\Delta_I} \ni Out^{\Delta_O}]$ *be standard metric ref/ret data constructed from either refinement data or onto retrenchment data for SysX and SysY.*

Let D_T, B_T *be positive constants. Let* $UI^T_{XY} \subseteq (\mathsf{U}_X \times \mathsf{U}_Y) \times \biguplus_{Op_X \geqslant Op_Y} (\mathsf{I}_{Op_X} \times \mathsf{I}_{Op_Y})$ *be a set of tuples of states and inputs of SysX and SysY such that:*

$$\begin{aligned}
&(u_X, u_Y, i_X, i_Y) \in UI^T_{XY} \Rightarrow \\
&\quad stp_{Op_X}(u_X, i_X, u'_X, o_X) \wedge stp_{Op_Y}(u_Y, i_Y, u'_Y, o_Y) \Rightarrow \\
&\quad d^{\ni}_{X,Y}(i_X, i_Y) \leq B_T \wedge d^G_{X,Y}(u_X, u_Y) \leq D_T \wedge \\
&\qquad (\exists\, \tilde{u}'_X \in \mathsf{U}_X, \tilde{o}_X \in \mathsf{O}_{Op_X} \bullet stp_{Op_X}(u_X, i_X, \tilde{u}'_X, \tilde{o}_X) \wedge \\
&\qquad\quad d^G_{X,Y}(\tilde{u}'_X, u'_Y) \leq D_T \wedge \\
&\qquad\quad EqEn^T_{X,Y}(\tilde{u}'_X, u'_Y))
\end{aligned} \tag{89}$$

where $EqEn^T_{X,Y}$ *is defined in (92) below. Let:*

$$\begin{aligned}
&EqEn^\tau_{X,Y}(u_X, u_Y) \equiv \\
&\quad [\ (\exists\, i_X \in \mathsf{I}_X, u'_X \in \mathsf{U}_X, o_X \in \mathsf{O}_X \bullet stp_{Op_X}(u_X, i_X, u'_X, o_X)) \Leftrightarrow \\
&\qquad (\exists\, i_Y \in \mathsf{I}_Y, u'_Y \in \mathsf{U}_Y, o_Y \in \mathsf{O}_Y \bullet stp_{Op_Y}(u_Y, i_Y, u'_Y, o_Y))\ ] \wedge \\
&\quad [\ stp_{Op_X}(u_X, i_X, u'_X, o_X) \wedge stp_{Op_Y}(u_Y, i_Y, u'_Y, o_Y) \Rightarrow (u_X, u_Y, i_X, i_Y) \in UI^T_{XY}\ ]
\end{aligned} \tag{90}$$

Let:

$$stp^T_Y \equiv \{\ stp_{Op_Y}(u_Y, i_Y, u'_Y, o_Y) \mid (\exists\, u_X \bullet EqEn^\tau_{X,Y}(u_X, u_Y))\ \} \tag{91}$$

Now, referring to Definition 8.15, let $\mathcal{K} \subseteq \mathbb{R}^+_{\neg 1}$ *be a finite set of positive constants, and for* $k \in \mathcal{K}$*, let* $B_k \in \mathbb{R}^+$ *be a positive constant, and:*

- *let* UI^{Tk}_{XY} *be as* UI^k_{XY} *is in Definition 8.15 but replacing* $EqEn_{X,Y}$ *by* $EqEn^T_{X,Y}$ *in (67),*
- *let* $EqEn^{Tk}_{X,Y}$ *be as* $EqEn^k_{X,Y}$ *is in Definition 8.15 but replacing* UI^k_{XY} *by* UI^{Tk}_{XY} *in (69),*
- *let* $stp^{Tk}_Y \equiv stp^k_Y - stp^T_Y$*, where* stp^k_Y *is given by (70).*

Let:

$$EqEn^T_{X,Y}(u_X, u_Y) \equiv EqEn^\tau_{X,Y}(u_X, u_Y) \vee \bigvee_{k \in \mathcal{K}} EqEn^k_{X,Y}(u_X, u_Y) \tag{92}$$

Suppose also that:

$$Init_X(u_X) \wedge Init_Y(u_Y) \Rightarrow G^{\Delta_G}_{X,Y}(u_X, u_Y) \wedge EqEn^T_{X,Y}(u_X, u_Y) \tag{93}$$

This collection of properties is called a threshold diverging/converging framework for SysX and SysY, where the stp^T_Y *components characterise the threshold aspect, the* stp^k_Y *components with* $1 < k \in \mathcal{K}$

characterise the diverging aspect, and the stp_Y^k components with $1 > k \in \mathcal{K}$ characterise the converging aspect.

Theorem 8.20 (Trace Inclusion with Threshold Diverging/Converging Framework). *Suppose given a threshold diverging/converging framework for SysX and SysY, assuming the notations of Definitions 8.15 and 8.19, where $\mathcal{K} = \{\kappa, K\}$, with $0 < \kappa < 1 < K$, and where $D_T = B_n(\Delta_G, n)$. Let* **M** *be a set of executions of SysY that satisfy the threshold* (M, n)-bounded excursion property *(where M and n are both constants), namely that, for every execution* **YY** *in* **M**:

(i) *All steps are in $stp_Y^T \cup stp_Y^\kappa \cup stp_Y^K$.*
(ii) *Every stp_Y^K step of* **YY** *is a member of a subsequence of at most M consecutive such steps.*
(iii) *Every stp_Y^κ step of* **YY** *is a member of a subsequence of at least n consecutive such steps, unless the subsequence is preceded or followed by at least one stp_Y^T step (in which case no length restriction applies).*

Let $In^B_{Op_{X,Y}}$ and $Out^B_{Op_{X,Y}}$ be given by (72)-(74), and let $G^B_{X,Y}$ be given by (84). Then:

(a) *If each trace in* **M** *starts with at least one stp_Y^K step, and M and n are related by (85), then there is an Init-constrained trace inclusion from the traces in* **M** *to traces of SysX, mediated by data $[G^B/In^B/Out^B]$ and Init-constraint $G^{\Delta_G}_{X,Y}$.*
(b) *If each trace in* **M** *starts with at least one stp_Y^T step or stp_Y^κ step, and M and n are related by (86), then there is a trace inclusion from the traces in* **M** *to traces of SysX, mediated by data $[G^{\Delta_G}/In^B/Out^B]$.*

The result holds if all the metric spaces involved do not have origins, or if they do, provided the relevant distance and relation symbols are interpreted correctly.

Proof. We reuse the proof of Theorem 8.16 to the greatest extent, arguing as follows. That proof can be interpreted as establishing the invariant that $d^G_{X,Y}$ never exceeded a value MAX, and that at the beginning of any sequence of stp_Y^K steps, it never exceeded a value MIN. The values MAX and MIN were different in the two cases (a) and (b) — in particular, the MIN value in case (b) was smaller than the MIN value in case (a).

In Theorem 8.16 the reduction in $d^G_{X,Y}$ needed to ensure that subsequent stp_Y^K steps respected the MAX bound was achieved by sequences of stp_Y^κ steps, which needed to be long enough to do the job. In the present situation, this is still the case, and for sequences of stp_Y^κ steps delimited by stp_Y^K steps, before and after, the same argument applies.

But if a sequence of stp_Y^κ steps is followed by a stp_Y^T step, the stp_Y^T step asserts that $d^G_{X,Y} \leq D_T = B_n(\Delta_G, n)$, which is less than the lower of the two MIN values of Theorem 8.16, and this holds regardless of the length of the sequence of stp_Y^κ steps, so the length restriction is not needed.

Likewise, if a sequence of stp_Y^κ steps is preceded by a stp_Y^T step, the stp_Y^T step asserts that $d^G_{X,Y} \leq D_T = B_n(\Delta_G, n)$, which is already small enough to ensure that the invariant is maintained, and subsequent stp_Y^κ steps can only diminish this value, so again, no length restriction is needed. The usual modularity in distance functions implies equal validity of no-origin and with-origin versions. □

Definition 8.21 (Unbounded Threshold Diverging/Converging Framework). *An unbounded threshold diverging/converging framework for SysX and SysY is defined as in Definition 8.19, but with the removal of the conjunct $(d^{\ni}_{X,Y}(i_X, i_Y) \leq B_k)$ from (the analogue of) (67), with the resulting term denoted $UI^{\circ Tk}_{XY}$. All terms derived directly or indirectly from $UI^{\circ Tk}_{XY}$ also acquire a $^\circ$ superscript thus: $EqEn^T_{X,Y} \rightarrow EqEn^{\circ T}_{X,Y}$, $EqEn^{Tk}_{X,Y} \rightarrow EqEn^{\circ Tk}_{X,Y}$, $stp_Y^{Tk} \rightarrow stp_Y^{\circ Tk}$, etc.*

Note that this definition does *not* remove the conjunct $(d^{\ni}_{X,Y}(i_X, i_Y) \leq B_T)$ from (89), constraining the behaviours of the stp^{τ}_{Y} steps in the same way as in Definition 8.19.

Theorem 8.22 (Simulation, Refinement, Retrenchment with (Unbounded) Threshold Diverging/Converging Framework). *Let $\mathcal{K} = \{\kappa, K\}$, with $0 < \kappa < 1 < K$. Let SysX and SysY be given, and let the following hold:* $\text{Ops}_X = \{Op\tau_X, Op\kappa_X, OpK_X\}$ *and* $\text{Ops}_Y = \{Op\tau_Y, Op\kappa_Y, OpK_Y\}$. *Assume the following retrenchment data:* $\succcurlyeq_{Ops_{X,Y}} = \{Op\tau_X \succcurlyeq Op\tau_Y, Op\kappa_X \succcurlyeq Op\kappa_Y, OpK_X \succcurlyeq OpK_Y\}$. *let:*

$$W^{\tau}_{Op\tau_{X,Y}} \equiv \text{true} \tag{94}$$

$$D^{\tau}_{Op\tau_{X,Y}} \equiv d^{G}_{X,Y}(u'_X, u'_Y) \leq d^{G}_{X,Y}(u_X, u_Y) \tag{95}$$

and for $k \in \mathcal{K}$, let $W^{k}_{Opk_{X,Y}}$ and $D^{k}_{Opk_{X,Y}}$ be as in Definition 8.17.

Suppose given an unbounded threshold diverging/converging framework for SysX and SysY, assuming the notations of Definitions 8.15 and 8.17. Assume (71) for the initial states. Then:

(a) *SysY is strongly comprehensively simulable by SysX, mediated by data* $[G^{\Delta_G}/\{W^{\tau}_{Op\tau}, W^{k}_{Opk}\}/\{D^{\tau}_{Op\tau}, D^{k}_{Opk}\}]$.

(b) *If the conjunct $(d^{\ni}_{X,Y}(i_X, i_Y) \leq B_k)$ is reinstated in the definition of $UI^{\circ Tk}_{XY}$ (so that the threshold diverging/converging framework is no longer unbounded), and the properties of M, n and G^B are assumed as in Theorem 8.20, then SysY is Init-constrained refinement simulable by SysX, mediated by data* $[G^B/\{W^{\tau}_{Op\tau}, W^{k}_{Opk}\}/\{D^{\tau}_{Op\tau}, D^{k}_{Opk}\}]$ *and Init-constraint G^{Δ_G}.*

(c) *Taking all possible SysX simulations $\boldsymbol{XX}$ of all possible SysY traces $\boldsymbol{YY}$ into account (as constructed for parts (a) and (b)), if* $[G^B/\{W^{\tau}_{Op\tau}, W^{k}_{Opk}\}/\{D^{\tau}_{Op\tau}, D^{k}_{Opk}\}]$ *are restricted to just those values that occur within these simulations, then there is a retrenchment from SysX to SysY with these restricted data.*

The result holds if all the metric spaces involved do not have origins, or if they do, provided the relevant distance and relation symbols are interpreted correctly.

Proof. This is almost identical to the proof of Theorem 8.18. Claim (a) follows identically. Claim (b) follows by the same modification that we used in Theorem 8.20 regarding growth of $d^{G}_{X,Y}$. Claim (c) follows identically. The usual no-/with-origin arguments apply. □

Theorems 8.16 and 8.20 (focused on trace inclusions for diverging/converging frameworks, without and with thresholds respectively) take a semantics directed approach in that operation names were not to the fore in the argument. For instance, the same operation (name) could be associated with transitions belonging to both stp^{K}_{Y} and stp^{κ}_{Y}. Theorems 8.18 and 8.22 however (as for the previous cases but with unboundedness), showed the opposite approach, where different operation names labelled conceptually distinct behaviours at the semantic level *ab initio*. Results such as these can serve as templates for a wide range of variants adapted to fit specific applications needs, based on considerations such as the following two, for example.

For a first, a wider variety of syntactic distinctions between categories of diverging and converging behaviours, each characterised by its own k constant, could easily be catered for, at the cost of a more complicated analysis. This has already been anticipated in the generality visible in Definitions 8.15, 8.17, 8.19, 8.21, which set up the machinery for the stated theorems.

For a second, different assumptions could be made about how inputs impact the changes of state for each category. In the results above, we assumed that inputs affected changes of state via a linear functional form, i.e. $d^{G}_{X,Y}(u'_X, u'_Y) \leq k(d^{G}_{X,Y}(u_X, u_Y) + d^{\ni}_{X,Y}(i_X, i_Y))$, but this need not be the only

useful form that could be considered. Again, there would be an impact on the complexity of the associated analysis.

9 CONTINUOUS BEHAVIOUR IN CONCEDING/RESTORING THEOREMS

The focus in the last few sections has been on before-after properties, treated in the traditional discrete transition style. As in Section 5, our aim in this paper is to cater for continuous behaviour via smooth adaptations of the discrete techniques. We thus reconsider the preceding results in this light.

The first point of call is Section 7. The results there were based purely on accessibility considerations; i.e. the existence of an after-state to which there is a transition (or a sequence of transitions) from some before-state playing a particular role in the discourse so far. The analogue of this for the case of continuous transitions is the existence of a continuous trajectory from the before-state to the (final) after-state (of the individual continuous transition or sequence of them). But this is already implicit in the hypothesis of there being one or more continuous transition(s) at all. Therefore, there is nothing more to be done for this situation, and we derive the following.

Proposition 9.1. *The results of Section 7 remain true if the systems involved include continuous transitions.*

Section 8 introduces metric arguments into the discourse. This raises the issue of how any change in metric properties of a quantity v that is acceptable regarding its before-state value 'v and after-state value v' in a discrete transition, should be interpreted over the interior of a continuous transition that has those before- and after-states at its ends $\mathbb{t}_{\mathrm{L}}$ and $\mathbb{t}_{\mathrm{R}}$, i.e. if $v(\mathbb{t}_{\mathrm{L}}) = $'$v$ and $v(\mathbb{t}_{\mathrm{R}}) = v$'. Given that the general aim of metric properties is to ensure that the magnitude of particular quantities does not become excessive, some natural interpretations suggest themselves:

Policies 9.2.

(1) *We demand that in the interior of a continuous transition the requisite magnitude is bounded by the larger of the two values at its ends:*

$$\mathbb{t}_{\mathrm{L}} \leq t \leq \mathbb{t}_{\mathrm{R}} \;\Rightarrow\; |v(t)| \leq \max\{|v(\mathbb{t}_{\mathrm{L}})|, |v(\mathbb{t}_{\mathrm{R}})|\} \tag{96}$$

(2) *We demand that in the interior of a continuous transition the requisite magnitude is bounded by the linear interpolation of the values at its ends:*

$$0 \leq \lambda \leq 1 \;\Rightarrow\; |v(\lambda\mathbb{t}_{\mathrm{L}} + (1-\lambda)\mathbb{t}_{\mathrm{R}})| \leq \lambda|v(\mathbb{t}_{\mathrm{L}})| + (1-\lambda)|v(\mathbb{t}_{\mathrm{R}})| \tag{97}$$

(3) *We demand that in the interior of a continuous transition the requisite magnitude is bounded by a monotonic function ϕ that is itself bounded between the smaller* MIN *and larger* MAX *of the values at its ends:*

$$0 \leq \lambda \leq 1 \;\Rightarrow\; \phi \text{ monotonic} \wedge \phi(\lambda) \in [\mathrm{MIN}\ldots\mathrm{MAX}] \wedge |v(\lambda\mathbb{t}_{\mathrm{L}} + (1-\lambda)\mathbb{t}_{\mathrm{R}})| \leq \phi(\lambda) \tag{98}$$

Evidently, Policy 9.2.(1) is the most lax, while Policy 9.2.(3) permits the definition of more finegrained continuous behaviour. Policy 9.2.(2) is a common particular case of Policy 9.2.(3).

Proposition 9.3. *If, in the interior of all pliant transitions, the metric of any quantity of interest in the discussions and results of Section 8 is a convex function of time, then it adheres to all of the Policies in Policies 9.2 (with suitable ϕ in the case of the third).*

Proof. Convexity is exactly the property in (97), and compliance with (97) implies compliance with (96). □

Of course, Proposition 9.3 merely delegates the proof of compliance with the stated properties, to proving convexity. Convexity considerations lie at the heart of many discussions of stability in control systems, a subject with a vast literature. We will assume in the rest of this paper that the pliant behaviour we need to deal with does in fact conform to the properties we need, as just described.

10 GRADED REFINEMENTS AND RETRENCHMENTS

We now extend the formal structures between a pairs of systems developed above to larger aggregations of system models.

10.1 Graded Development Systems

A **graded development system** (GDS) is built out of the ingredients described in the preceding sections. First, there is a strictly partially ordered finite set of system labels $\mathfrak{X} = \{X, Y, Z, \ldots\}$. We use $\succcurlyeq$ for the partial order, e.g. $X \succcurlyeq Y$. For each label X there is a system $SysX$, and for each case of $X \succcurlyeq Z$ (which we will call a link) there is: *either* a retrenchment relationship $SysX \succcurlyeq SysZ$ from $SysX$ to $SysZ$, witnessed by relevant pairs of operations $\mathrm{Ops}_X \succcurlyeq_{\mathrm{Ops}_{X,Z}} \mathrm{Ops}_Z$, themselves witnessed by data $G_{X,Z}, W_{OpX,Z}, D_{OpX,Z}$ in each case; *or* there is merely a suite of retrenchment data, as just described. We use the same terminology when the pairs of operations are replaced by (m, n) diagrams instead.

Because $\mathfrak{X}$ is finite, there will be a unique *covering* subrelation $\succcurlyeq_\bullet$ of $\succcurlyeq$ consisting of links $X \succcurlyeq Z$ such that there is no Y such that $X \succcurlyeq^+ Y \succcurlyeq^+ Z$. We call the links of $\succcurlyeq_\bullet$ basic links.

We know that the composition of retrenchment data $W_{Op_{X,Y}}, D_{Op_{X,Y}}$, discussed in Section 4.1, is associative (because it is just based on composition of relations). This enables paths in $\mathfrak{X}$ e.g. $X \succcurlyeq^+ Z \equiv X \succcurlyeq Y \succcurlyeq Z$ (which are constructed by transitive closure of $\succcurlyeq_\bullet$) to be mapped to retrenchment data relating the systems at the paths' ends by the composition of the data belonging to the basic links involved, e.g.:

$$SysX \succcurlyeq_{[G_{X,Z}/W_{X,Z}/D_{X,Z}]} SysZ \;\equiv\; SysX \succcurlyeq_{[G_{X,Y}\fatsemi G_{Y,Z}/W_{X,Y}\fatsemi W_{Y,Z}/D_{X,Y}\fatsemi D_{Y,Z}]} SysZ \tag{99}$$

If, for all basic links $X \succcurlyeq_\bullet Z$ we have unique retrenchment data $SysX \succcurlyeq_{[G_{X,Z},W_{OpX,Z},D_{OpX,Z}]} SysZ$, and all other retrenchment data between the systems labelled by $\mathfrak{X}$ arise by composition of the basic link data for these (in all possible ways), then we say the GDS is a *simple* GDS (SGDS). Thus for a SGDS, there is a total function from non-empty paths in $\mathfrak{X}$ to retrenchment data between the systems at the paths' ends, which, for paths consisting of more than one basic link, arise by composition. We extend this to include the case $X = Z$ by insisting that the relevant retrenchment is the identity (given by identity relations for G, W and D), thus extending to reflexive transitive closure, and the maximally non-strict extension of $\succcurlyeq$. The same applies if the retrenchment data witness actual retrenchments between the systems concerned.

Looking at all that from a categorical perspective, the projection that maps each system $SysX$ of an SGDS to its label X, and maps each retrenchment (data) $SysX \succcurlyeq SysY$ to the link $X \succcurlyeq Y$ is a(n op)fibration, split in fact [23, 25, 54].

In general, if there is more than one path from X to Z in $\mathfrak{X}$, the retrenchment data belonging to those paths may be combined using the techniques discussed in Section 4.1 (although if arbitrary combinations of this kind are included in the collection of retrenchments associated with $\mathfrak{X}$, the fibration properties just mentioned will be impaired).

Let $\gamma = X \succcurlyeq Y \succcurlyeq \ldots \succcurlyeq Z$ be a directed path from X to Z in $\mathfrak{X}$. We write $\overline{\gamma} = Z \preccurlyeq \ldots \preccurlyeq Y \preccurlyeq X$ for the dual path, i.e. the same thing regarded as a path from Z to X instead. Given such a path $\gamma = X \succcurlyeq Y \succcurlyeq \ldots \succcurlyeq Z$, we can write $[G_\gamma, W_\gamma, D_\gamma]$ for the retrenchment data derived by composing the data of the constituent links.

Now let $\gamma = X \succcurlyeq Y_1 \preccurlyeq Y_2 \succcurlyeq Y_3 \ldots \succcurlyeq Z$ be a not-unidirectional path (NUD path) from X to Z in $\mathfrak{X}$, i.e., not all links point in the same direction. We can extend the previous definition of $[G_\gamma, W_\gamma, D_\gamma]$ for the retrenchment data for such a path, by using, in the composition of the data belonging to the links, the transposes of the relations for the data belonging to 'wrongly oriented' links. Clearly, for NUD paths, there is no conceptual difference between a path and its dual.

With the above understood, given γ, we define V_γ and $V_{\overline{\gamma}}$ by:

$$V_\gamma \subseteq \cup_Z \;\equiv\; \{\, z \in \cup_Z \mid \neg(\exists x \bullet G_\gamma(x,z)) \,\} \tag{100}$$

$$V_{\overline{\gamma}} \subseteq \cup_X \;\equiv\; \{\, x \in \cup_X \mid \neg(\exists z \bullet G_\gamma(x,z)) \,\} \tag{101}$$

Similarly, we define $VInv_\gamma$ and $VInv_{\overline{\gamma}}$ by:

$$VInv_\gamma \subseteq \cup_Z \;\equiv\; \{\, z \in \cup_Z \mid Inv_Z(z) \wedge \neg(\exists x \bullet Inv_X(x) \wedge G_\gamma(x,z)) \,\} \tag{102}$$

$$VInv_{\overline{\gamma}} \subseteq \cup_X \;\equiv\; \{\, x \in \cup_X \mid Inv_X(x) \wedge \neg(\exists z \bullet G_\gamma(x,z) \wedge Inv_Z(z)) \,\} \tag{103}$$

Referring back to Section 2, we can make analogous definitions $VCInv_\gamma, VCInv_{\overline{\gamma}}$ by replacing occurrences of Inv in (102)-(103) with corresponding instances of the contingent invariant concept $CInv$. Likewise, we can strengthen all these concepts by insisting on reachability, or reachability alongside invariance or contingent invariance, thereby getting $VReach_\gamma, VReach_{\overline{\gamma}}$ and $VReachInv_\gamma, VReachInv_{\overline{\gamma}}$ and $VReachCInv_\gamma, VReachCInv_{\overline{\gamma}}$. There will also be a vast array of mixed quantities, demanding different properties at the two ends (and perhaps at intermediate points) of γ, which we do not list.

Since, between two models of interest to the same application, G_γ is likely to be simple and to just reflect some obvious structural relationship between the state spaces, $V_\gamma, V_{\overline{\gamma}}$ are unlikely to be informative. However, the situation changes for the other quantities defined.

If $X \succcurlyeq^+ Z$, then, speaking intuitively, we regard $SysX$ as 'abstract' and $SysZ$ as 'concrete'. Then, from a system engineering point of view, states in $VInv_\gamma$ are 'concrete' states, potentially reachable in an implementation level model, that do not have an 'abstract' counterpart. This can happen if an abstract model is too idealised compared with the detailed concrete model, for reasons of simplicity or perspicuity at the abstract level. In such situations, $\succcurlyeq$ indicates the direction towards the introduction of non-ideal, implementation level detail, and the non-emptiness of $VInv_\gamma$ is not, of itself, problematic.

Conversely, states in $VInv_{\overline{\gamma}}$ are abstract states that do not have a counterpart in a more concrete, more implementable model. Since progress towards implementation using refinement is often accompanied by a narrowing of possibilities (e.g. the often quoted 'reduction of nondeterminism' slogan), this, in itself, is not problematic. An exception arises when the abstract states (and their associated system behaviour) are abstract representations of essential system requirements.[14] In that case, non-emptiness of $VInv_{\overline{\gamma}}$ could indicate non-fulfillment of these requirements as implementation is approached, and such a state of affairs would need to be properly evaluated within the development process. Thus, if the abstract representations of the system requirements are, in essence, accurate, then their non-fulfillment needs to be reconciled with a more accurate requirements model. On the other hand, if the abstract representations of the requirements are merely over-idealised, then the situation is similar to the unproblematic case described earlier.

The situation becomes more challenging to interpret when we deal with NUD paths. For NUD paths where the links are predominantly in one direction, the abstract to concrete interpretation *vis a vis* requirements can be maintained, provided the oppositely oriented links are associated with transposes of the original relations. But for NUD paths which offer a much more balanced

[14] Event-B [3] advocates the stepwise incorporation of requirements via successive refinements. The ASM technique [26, 27] also speaks of refinement toward the 'ground model' which incorporates *all* the requirements.

mix of forward and backward links, such a view is harder to sustain. We take the view that such situations are best dealt with on a case by case basis.

10.2 Graded Development Systems with Approximations

Thus far, $\mathcal{X}$ is an arbitrary (finite partially ordered) set. Now we choose to specialise $\mathcal{X}$ somewhat. Let $\mathcal{A} = \{A, B, \ldots\}$ be a finite set (strictly partially ordered by $\succcurlyeq$) of (names for) abstraction levels. Let ordr map $\mathcal{A}$ to the naturals, ordr : $\mathcal{A} \rightarrow \mathbb{N}$. We stipulate that $\mathcal{X}$ is of the form:

$$\mathcal{X} = \mathcal{A} \cup \{(A, k) \mid A \in \mathcal{A} \ \wedge \ 0 < k \leq \mathrm{ordr}(A)\} \tag{104}$$

The strict partial order $\succcurlyeq$ on such an $\mathcal{X}$ is now defined with the help of the usual lexicographical ordering:

$$A \succcurlyeq B \qquad \text{if} \quad \mathrm{ordr}(A) = \mathrm{ordr}(B) = 0 \ (\text{and } A \succcurlyeq B \text{ in } \mathcal{A}) \tag{105}$$

$$A \succcurlyeq (B, n) \qquad \text{if} \quad A \succcurlyeq B \text{ in } \mathcal{A} \wedge \mathrm{ordr}(A) = 0 \tag{106}$$

$$(A, n) \succcurlyeq B \qquad \text{if} \quad A \succcurlyeq B \text{ in } \mathcal{A} \wedge \mathrm{ordr}(B) = 0 \tag{107}$$

$$A \succcurlyeq (A, n) \qquad \text{if} \quad 0 < n \tag{108}$$

$$(A, n) \succcurlyeq (B, m) \qquad \text{if} \quad A \succcurlyeq B \ \vee \ (A = B \ \wedge \ 0 < n < m) \tag{109}$$

The idea behind this is that the elements of $\mathcal{A}$ can index major levels of modelling or design abstraction, while the integer part of a label (A, n) is available to indicate the order of approximation in $Sys(A, n)$, on the assumption that $Sys(A, n)$ represents an n'th order approximation to an ideal level A model which is converged to ever more closely as n increases, assuming in turn, that the ideal model itself does not allow convenient closed form expression (or even if it does, when computations with the closed form would need to resort to numerical approximation).[15]

Allowing the order of approximation for a given level A to be limited by $\mathrm{ordr}(A)$, and allowing the partial order on $\mathcal{X}$ to be a subset of the full lexicographical ordering on all (A, k) pairs permits us to focus on what we regard as the important relationships in the development structure.

In a figurative sense, we can view a position higher in the partial order on $\mathcal{X}$ as labelling a system model that is 'more ideal' than one in a position lower down. As observed before, 'ideal' can both indicate a better expression of the system requirements, and also a less realistic expression of them.

Our definition also embodies, in (108), the view that any approximation lives lower down in the $\succcurlyeq$ order than a mathematically more ideal model that it approximates (given perhaps, implicitly as a solution to an ODE system, or using infinite series, etc.). This also extends to the view, in (109), that a higher order approximation, lives lower down in the $\succcurlyeq$ order than a lower order one — this being because a higher order approximation entails greater low level complexity and is thus closer to implementation.

By contrast, the alternative view, that a higher order approximation is going to be metrically closer to the ideal model it approximates, so should be higher in the $\succcurlyeq$ order, is also justifiable — in the end, it is a matter of choice. Such a state of affairs would be captured by changing the inequality in the right hand side of (109) to $0 < m < n$. In this paper we stick to the former view.

In considering systems up to approximation, relations that are themselves approximate are often used. In mathematics it is conventional to write, for example, $A * B + \mathrm{o}(t^2)$, where $*$ is some suitable relation. In this paper we will typically write this in infix form as:

$$A \ *^{\mathrm{o}(t^2)} \ B \tag{110}$$

to maintain conformance with other relational notations.

[15] The idea of allowing approximations, (and specifically to n'th order and similarly) to appear in formal development schemes, has been considered before, although the approaches seen in the literature differ from ours. See e.g. [24].

MACHINE *HyEvBMch*
TIME t
CLOCK clk
PLIANT x, y
VARIABLES u
INVARIANTS
$x, y, u \in \mathbb{R}, \mathbb{R}, \mathbb{N}$
EVENTS
INITIALISATION
STATUS ordinary
WHEN
$t = 0$
THEN
$clk := 1$
$x, y, u := x_0, y_0, u_0$
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
$\mathcal{D}\, x = \phi(x, y, u, i?, l, o!, t, clk)$
$y, o! := E(x, u, i?, l, t, clk)$
END
END

Fig. 2. A schematic Hybrid Event-B machine.

11 HYBRID EVENT-B OVERVIEW

In this section we embark on the presentation of our motivating case study. This is developed in the Hybrid Event-B (HEB) formalism [12, 13], so we start by giving an overview of this. Note that HEB is a syntactic formalism, in contrast to the semantic approach of the previous sections. We make suitable comments where necessary.

In Fig. 2 we see a skeletal 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 and 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 (i.e. one unit per unit of time), but they may be set during mode events (see below).

Variables are of two kinds. There are mode variables (like u, declared in the usual manner) 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; these changes are specified via pliant events (see below). 'Pliant' is the shorter word used in HEB to refer to continuous concepts.

Next are the invariants. These resemble invariants in discrete Event-B [3], in that the types of the variables are asserted to be the (static) sets from which the variables' values *at any given moment of time* are drawn. More complex invariants are similarly predicates involving any or all of the variables that are required to hold *at all moments of time* during a run.

The events start with *INITIALISATION*, which has a guard that synchronises time with the start of any run (the WHEN clause), while all other variables are assigned their initial values in the usual way (in the THEN clause that complements the WHEN clause). 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, and defines mode event enabledness. 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. The usual abbreviations using assignment notation such as := are available.

Pliant events are exclusive to HEB. They specify the continuous evolution of the pliant variables over an interval of time.[16] 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. Their conjunction defines pliant event enabledness. The separation between the two guards is motivated by considerations connected with refinement (discussed in detail in [12]).

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 using two specification mechanisms: direct assignments and ordinary differential equations (ODEs). For example, the behaviour of pliant variable y and output variable $o!$ is given by a direct assignment to the (time dependent) value of the (vector valued) expression E, in $y, o! := E(\ldots)$. By contrast, the behaviour of pliant variable x is given by the solution to the first order ODE $\mathcal{D}\, x = \phi(\ldots)$, where $\mathcal{D}$ indicates differentiation with respect to time. (In fact the semantics of the $y, o! := E$ case can be given (modulo some technicalities concerning discontinuities) in terms of the ODE $\mathcal{D}\, y, \mathcal{D}\, o! = \mathcal{D}\, E$, so that x, y and $o!$ 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 machine's plaint events. (Often these are refined to more deterministic behaviour at lower levels of abstraction.) The COMPLY clauses of the relevant pliant events can house such constraints directly, leaving it to lower level refinements to add the necessary details of the dynamics.

If, from Fig. 2, we erase time, clocks, pliant variables and pliant events, we arrive at a skeleton (conventional) Event-B machine. This simple erasure process illustrates (in reverse) the way that HEB has been designed as a clean extension of the original Event-B framework [3]. The only difference of note is that, now —at least according to the (conventional) way that Event-B is interpreted in the physical world— (the mode) events (left behind by the erasure) execute *lazily*, i.e. *not* at the instant they become enabled (which is, of course, the moment of execution of the previous mode event).

Briefly, the semantics of a HEB machine 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. A run starts at some initial moment of time t_0, and lasts either for a finite time, or indefinitely. The duration of the run, $\mathcal{T}$, an interval of the reals, breaks up into a succession of left-closed right-open subintervals: $\mathcal{T} = [t_0 \ldots t_1), [t_1 \ldots t_2), [t_2 \ldots t_3), \ldots$, exactly as described in Section 5.

[16] In HEB terminology, a 'pliant event' syntactically specifies a 'continuous transition' as used earlier in the paper.

Mode events take place at the isolated times corresponding to the common endpoints of these subintervals t_i. In between, the mode variables are constant, and the pliant events stipulate continuous change in the pliant variables. We insist that on every subinterval $[t_i \ldots t_{i+1})$ the behaviour is governed by a well posed initial value problem [76]. Time t_{i+1} is defined as the earliest time at which a mode event becomes enabled, at which point the continuous behaviour is preempted, the mode event executes, and a further pliant event is executed after its completion. A system run is *well formed*, and thus belongs to the semantics of the machine, provided that at runtime:

(1) 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).[17]
(2) Every enabled pliant event is feasible, i.e. has a time-indexed family of after-states, and EITHER:
  (a) During the run of the pliant event a mode event becomes enabled. It preempts the pliant event, defining its end. ORELSE
  (b) During the run of the pliant event it becomes infeasible: finite termination. ORELSE
  (c) The pliant event continues indefinitely: nontermination.

Thus, in a well formed run, mode events alternate with pliant events.

Of course, there are many semantic details of the linguistic framework just described that are glossed over by the above description. These are covered with precision in [12, 13]. Nevertheless, we highlight the following points.

Clearly, the 'raw' state space of a HEB machine is the Cartesian product of the types of its variables. For this reason, the invariants, and the events' guards play a much more prominent role is shaping the accessible state subspace to the needs of the intended application than when we are at liberty to postulate it semantically.

Moreover, for a given machine, all the properties that we have been insisting hold in the account above, get translated to *proof obligations* (POs). These are theorem schemas that have been instantiated using elements extracted from the machine definition, and that must be shown to be true if the system model is to be regarded as correct.

Among the most prominent of the POs are ones that directly reflect the invariant preservation properties discussed in Section 2 (and their counterparts for pliant events discussed in Section 5); also the POs that ensure well formedness, discussed just above.

One class of POs not derived from the earlier discussion concerns *feasibility*, i.e. the existence of things that are needed, but that are specified purely syntactically. For example we can mention existence of initial states: specifically, if initial states are specified using a predicate expression *Init*, is this predicate expression actually satisfied by any state value? We can also mention existence of after-states (or of after-state families for the pliant case) of events that are specified syntactically; also, existence of suitable abstract states or transitions in refinement or retrenchment properties.

The latter point raises the issue of refinement and retrenchment of HEB machines. We will have a lot more to say about these in the context of our case study, so we postpone further discussion to Section 13.

12 ACTIVE CONTROL FOR EARTHQUAKE PROTECTION

An active control system for earthquake protection of a building is, like almost all control systems, steeped in considerations arising from conventional applied mathematics. In [11] the authors explore a formal development of such a system using a formal approach based on HEB. This enabled the many application level calculations needed, to be related directly to formal elements.

[17] If a mode event has an input, the semantics assumes that its value only arrives at a time strictly later than the previous mode event, ensuring part of 1 and 2 automatically.

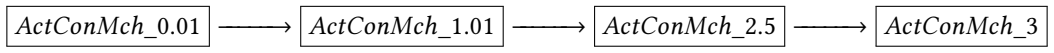

Fig. 3. An earthquake damage prevention active control system development hierarchy.

In this section we present the essentials of this development by describing a series of HEB models, shown in Fig. 3.[18] In the next section we pick up on the relationships between these models, and the subtleties related to these relationships, and thereby we derive a version of Fig. 3 in which the arrows are more informative.

In brief, a simple, single degree of freedom model of an active control system for a building, is based on the following second order ODE:

$$m\,\mathcal{D}^2\,x + c\,\mathcal{D}\,x + k\,x = p - m\,e? \tag{111}$$

In (111), m is the mass of the building and x is the displacement of a fiducial point in the building from where it should normally be (in an inertial frame of reference). In the active control mechanism, c is the coefficient of the viscous damper, and k is the spring constant of the mechanism. The force applied by the control mechanism is p, and $e?$ is the acceleration of the earth during an earthquake. This forms the focus of the development.

In Fig. 4 we see *ActConMch*_0.01, the top level HEB machine of the development. It does very little. Aside from the *INITIALISATION*, the only event, the pliant *MONITOR*, merely demands that provided that the environment's input $e?$ does not exceed the bound E_B in magnitude, then the *INVARIANTS* are maintained, the only non-trivial one of which states that the building displacement x stays within the safe margin X_B in magnitude. This constitutes a paradigmatic example of an abstract model merely expressing a system requirement, without concerning itself with the means by which the requirement is to be met.

Fig. 5 contains the code for *ActConMch*_1.01, a first concretisation of *ActConMch*_0.01. The behaviour of the *MONITOR* event is a constraining of its previous incarnation. As well as the input $e?$, there are now two locally chosen parameters, pp and e. The former, via the assignment $p := pp$, allows values that match $e?$ only imprecisely, to be fed to the ODE system in the SOLVE clause, 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 SOLVE clause itself decomposes

[18] See [11] for a more detailed discussion than appears here.

```
MACHINE ActConMch_0.01
PLIANT x
INVARIANTS
  x ∈ ℝ
  |x| ≤ X_B
EVENTS
  INITIALISATION
    STATUS ordinary
    BEGIN
      x := 0
    END
... ...
```

```
... ...
  MONITOR
    STATUS pliant
    ANY e?
    WHERE
      e? ∈ ℝ ∧ |e?| ≤ E_B
    COMPLY INVARIANTS
    END
END
```

Fig. 4. A highly abstract model of the earthquake damage prevention active control system: *ActConMch*_0.

MACHINE *ActConMch*_1.01
CLOCK *clk_pls*
PLIANT x, y, p
INVARIANTS
$x, y, p \in \mathbb{R}, \mathbb{R}, \mathbb{R}$
$|x| \leq X_B$
EVENTS
INITIALISATION
STATUS ordinary
BEGIN
$clk\_pls := 0$
$x, y, p := 0, 0, 0$
END
MoSkip
STATUS ordinary
WHEN
$clk\_pls = T_P$
THEN
$clk\_pls := 0$
END
. . . . . .

. . . . . .
MONITOR
STATUS pliant
ANY $pp, e, e?$
WHERE
$pp \in \mathbb{R} \wedge |pp| \leq PP_B \wedge$
$e \in \mathbb{R} \wedge \mathrm{CONST}(e) \wedge$
$e? \in \mathbb{R} \wedge |e?| \leq E_B \wedge |e - e?| \leq e_B$
COMPLY
$\{e^{\boldsymbol{A}(t-\mathbb{t}_L)}[x(\mathbb{t}_L), y(\mathbb{t}_L)]^T +$
$(e^{\boldsymbol{A}s} *_{[\mathbb{t}_L \ldots t]} [0, \frac{1}{m}pp - e?]^T)\}.x$
$\leq X_B$
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. 5. A more concrete model of the system: *ActConMch*_1.

the second order ODE (111) into two first order ODEs, one for the displacement x and one for the velocity y.

The imprecision between pp and $e?$ is certainly needed, since it is infeasible to require that p be chosen to exactly match $me?$ in real time. The latter would enable the identically zero solution to (111), enabling the building to stand perfectly still in even the most violent earthquake. Some bound like e is also needed since the any engineered system has finite capability, and cannot be expected to perform outside its limits.

The COMPLY clause takes advantage of the fact that the solution to linear constant coefficient inhomogeneous ODE systems such as (111) is routine. See [4, 29, 72, 76] as well as a host of other sources. The first term is the homogeneous solution, primed by the initial values: $e^{\boldsymbol{A}(t-\mathbb{t}_L)}[x(\mathbb{t}_L), y(\mathbb{t}_L)]^T$, where $\boldsymbol{A}$ is the companion matrix of the homogeneous part of the ODE system in the SOLVE clause, and $\mathbb{t}_L$ refers, generically, to the start time of any runtime transition specified by the pliant event. The second term is the convolution $*$ over the interval $[\mathbb{t}_L \ldots t]$ between the homogeneous solution $e^{\boldsymbol{A}(s)}$ (with bound convolution variable renamed to s) and the inhomogeneous part $[0, \frac{1}{m}pp - e?]^T$. 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 $e?$ makes this a practical proposition.

Owing to the introduction of the clock *clk_pls*, the mode event *MoSkip* interrupts *MONITOR* 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.

In Figs. 6-7 there are two further stages of the development. The text in the figures that is purely black constitutes machine *ActContMch*_2.5. Adding the parts in red (of the form $K^{-1}_{\ldots} \lfloor K_{\ldots} \ldots \rceil$)

gives machine *ActContMch_3*. Machine *ActContMch_2.5* adds discrete time data sampling (to the machine model *ActContMch_2* discussed in [11]), while *ActContMch_3* adds sensor and actuator quantisation, via the $K_{\ldots}^{-1}\,\lfloor K_{\ldots}\ldots\rceil$ additions, which stipulate scaling, rounding, and unscaling.

Looking more closely at the details of *ActContMch_2.5*, compared with its predecessor, x and y have become xx and yy for easy discrimination later. The needed invariant becomes $|xx| \leq X_B$.

The pliant behaviour of *MONITOR* is interrupted every T_P time units by the *PulseXX* mode events. These have the capability of applying an impulse to the building if the ground motion is such that damage to the building is threatened.

PulseNo (which takes no action) is executed if the ground motion is below a safe threshold X_{th}. *PulseMaybe* is executed if the ground motion is above the X_{th} threshold, but a more detailed calculation shows that action is, in fact, not needed during the current sampling interval. The detailed calculation is given by $\left| x19\,(1-\omega^2\,T_P^2/2) + y19\,T_P(1-\zeta\,\omega\,T_P) - e19\,T_P^2/2 \right| \leq X_B$, in which $\zeta = c/2\sqrt{km}$ and $\omega = \sqrt{k/m}$. These constants are derived from a standardisation of the LHS of the ODE (111) into the form $\mathcal{D}^2\,x + 2\zeta\omega\,\mathcal{D}\,x + \omega^2\,x$. See [11, 30] for further details.

PulseYesY and *PulseYesE* cover the cases when the inequality just quoted is reversed — in the former case the contribution from the yy variable to the excess is greater, so a pulse is delivered that reverses its effect, in the latter case the contribution from ground motion to the excess is greater, so its effect is counteracted by reversing that.

```
MACHINE ActContMch_2.5/3
CLOCK clk_pls
VARIABLES
  x18, x19, y19, e19
PLIANT xx, yy
INVARIANTS
  x18, x19, y19, e19 ∈ ℝ, ℝ, ℝ, ℝ
  xx, yy ∈ ℝ, ℝ
  |xx| ≤ X_B
EVENTS
  INITIALISATION
    STATUS ordinary
    BEGIN
      clk_pls := 0
      x19, y19, e19 := 0, 0, 0
      xx, yy := 0, 0
    END
  MONITOR
    STATUS pliant
    ANY e, e?
    WHERE
      e ∈ ℝ ∧ CONST(e) ∧
      e? ∈ ℝ ∧ |e?| ≤ E_B ∧ |e − e?| ≤ e_B
    SOLVE
      𝒟xx = yy
      𝒟yy = −(c/m) yy − (k/m) xx − e?
    END
... ...
```

```
... ...
  Sample_18
    WHEN
      clk_pls = (18/20) T_P
    THEN
      x18 := K_xs^-1 ⌊K_xs xx⌉
    END
  Sample_19
    ANY e?
    WHERE
      clk_pls = (19/20) T_P ∧
      e? ∈ ℝ ∧ |e?| ≤ E_B
    THEN
      x19 := K_xs^-1 ⌊K_xs xx⌉
      y19 := (K_xs^-1 ⌊K_xs xx⌉ − x18) 20/T_P
      e19 := K_es^-1 ⌊K_es e?⌉
    END
  PulseNo
    STATUS ordinary
    WHEN
      clk_pls = T_P ∧ |x19| < X_th
    THEN
      clk_pls := 0
    END
... ...
```

Fig. 6. The active control system, versions *ActContMch_2.5* and *ActContMch_3*, first part.

```
... ...
    PulseMaybe
      STATUS ordinary
      WHEN
        clk_pls = T_P ∧ |x19| ≥ X_th ∧
        | x19 (1 − ω² T_P²/2) + y19 T_P(1 − ζ ω T_P) − e19 T_P²/2 | ≤ X_B
      THEN
        clk_pls := 0
      END
    PulseYesY
      STATUS ordinary
      ANY Δx, w
      WHERE
        clk_pls = T_P ∧ |x19| ≥ X_th ∧
        | x19 (1 − ω² T_P²/2) + y19 T_P(1 − ζ ω T_P) − e19 T_P²/2 | − X_B = w ∧
        w > 0 ∧
        Δx = w + (X_B − |x19|) ∧
        [(sign(y19) = sign(−e19) ∧ |y19 T_P(1 − ζ ω T_P)| ≥ Δx/2) ∨
         (sign(y19) ≠ sign(−e19) ∧ |y19 T_P(1 − ζ ω T_P)| ≥ Δx)]
      THEN
        clk_pls := 0
        yy := K_ys^−1 ⌊K_ys (−y19)⌉
      END
    PulseYesE
      STATUS ordinary
      ANY Δx, w
      WHERE
        clk_pls = T_P ∧ |x| ≥ X_th ∧
        | x19 (1 − ω² T_P²/2) + y19 T_P(1 − ζ ω T_P) − e19 T_P²/2 | − X_B = w ∧
        w > 0 ∧
        Δx = w + (X_B − |x19|) ∧
        [(sign(y19) = sign(−e19) ∧ | − e19 T_P²/2| ≥ Δx/2) ∨
         (sign(y19) ≠ sign(−e19) ∧ | − e19 T_P²/2| ≥ Δx)]
      THEN
        clk_pls := 0
        yy := K_ys^−1 ⌊K_ys (e19 T_P/2 (1 − ζ ω T_P))⌉
      END
  END
```

Fig. 7. The active control system, versions *ActContMch_2.5* and *ActContMch_3*, second part.

Of course, all these mode events need values for xx and yy, and these are supplied by the *Sample_18* and *Sample_19* events, which sample the xx variable at $\frac{18}{20}T_P$ and $\frac{19}{20}T_P$, and use the two xx values to estimate the velocity yy. The earth movement is also sampled, yielding altogether the values in variables $x18, x19, y19, e19$. These remarks make it clear that the functioning of machine *ActContMch_2.5* is discretised in time, albeit that its discrete actions are determined by the continuous behaviour.

Adding the $K_{\ldots}^{-1}\lfloor K_{\ldots}\ldots\rceil$ parts gives machine *ActContMch_3*. The K values being constants, these parts implement scaling, rounding and unscaling, and play the role of surrogates for signal

quantisation, whereby sensor and actuator values are determined digitally and thus take one of a finite number of values, depending on the device in question.

A detailed discussion of the various constants and values appearing in Figs. 6-7 can be found in [11]. For us, the salient point is that the introduction of the surrogate quantisation around the otherwise smooth evolution of the xx and yy variables is guaranteed to provoke the kind of threshold crossing problems noted in the Introduction when moving from machine *ActContMch*_2.5 to machine *ActContMch*_3. It is also clear that the discretisation present in both machines would introduce similar problems if compared with a machine in which the values needed for the pulse calculations were extracted instantaneously at the boundaries of T_P intervals (as in machine *ActContMch*_2 of [11]). Since the calculations for that would be more complicated, we avoid considering them in this paper — the effects introduced by the K values are sufficient for us, and the differences in the models being compared are easy to focus on in machines *ActContMch*_2.5 and *ActContMch*_3.

We close this presentation of the machines by noting that, strictly speaking, none of them appears in [11]. Machines *ActConMch*_0.01 and *ActConMch*_1.01 differ from the corresponding machines of [11] in minor technical detail (connected with the fact that we are not using the syntactically based HEB definition of refinement here, but the semantically based one of this paper). Also, in [11], the account jumps from *ActContMch*_2, which is a single machine encapsulating the system behaviour in mathematically idealised terms, to *ActContMch*_4, which is a multi-machine system capturing the distributed nature of the real system, and incorporating the quantisation effects.

13 RELATING THE VARIOUS EARTHQUAKE PROTECTION MODELS

In Section 12 the four models of Fig. 3 were discussed in isolation. We now address this omission, using the refinement and retrenchment notions developed earlier in this paper, and commenting on minor detailed difference from the corresponding HEB and ASM notions where this is helpful. We use subscripts 0, 1, 2, 3 to distinguish quantities that occur in more than one of the models, as needed.

13.1 *ActContMch*_0.01 and *ActContMch*_1.01

To start with, we claim that *ActContMch*_1.01 is a retrenchment of *ActContMch*_0.01. This is fairly clear when we observe that the trajectories for x permitted by *ActContMch*_1.01 are a subset of those permitted by *ActContMch*_0.01. The trajectories in *ActContMch*_1.01 are specified in a more constrained way than those in *ActContMch*_0.01, and in addition, there is no constraint in *ActContMch*_0.01 that is not also required by *ActContMch*_1.01. It all works provided we relate the two models using mainly identities and projections, and provided the many constants playing a part in the two machines are suitably chosen (so that the sets of trajectories in the two cases are nonempty). We assume this to be the case.

Thus we adopt the following retrenchment data. Regarding the events, we have $\succcurlyeq \equiv \{MONITOR_0 \succcurlyeq MONITOR_1\}$. The gluing relation between state spaces, $G_{0,1} : \mathsf{U}_0 \leftrightarrow \mathsf{U}_1$, is given by the converse of the projection of (x, y, p, clk_pls) tuples to x values alone. The within relation $W_{MONITOR_{0,1}} : \mathsf{U}_0 \times \mathsf{I}_{MONITOR_0} \leftrightarrow \mathsf{U}_1 \times \mathsf{I}_{MONITOR_1}$ is the converse of the projection of $(x, y, p, clk\_pls, pp, e, e?)$ tuples to $(x, e?)$ tuples. The delivers relation $D_{MONITOR_{0,1}} : \mathsf{U}_0 \times \mathsf{I}_{MONITOR_0} \leftrightarrow \mathsf{U}_1 \times \mathsf{I}_{MONITOR_1}$ is the same as $W_{MONITOR_{0,1}}$ as there are no outputs (and we do not single out initial state values).

To substantiate the claim that the given retrenchment data support an actual retrenchment, we much check the various conditions for a retrenchment. Firstly, the initialisation (5). This is trivial since $Init_0$ and $Init_1$ are identical on the common variable x, and $G_{0,1}$ is an identity on x. Next, the correctness PO for $MONITOR_0$ and $MONITOR_1$. This demands that $if\,(W_{MONITOR_{0,1}} \wedge MONITOR_1)$

holds over a given time interval, *then* ($MONITOR_0 \wedge D_{MONITOR_{0,1}}$) does too, over the same interval. But since we can choose $MONITOR_0 = MONITOR_1$, and $D = W$, this is trivial again.

The fact of all concrete transitions projecting to abstract ones typifies exemplary behaviour for a refinement. Why then the claim that we only have a retrenchment? The culprit is $\succcurlyeq$ on events, which is not onto the mode event *MoSkip*. This event preempts $MONITOR_1$, although it does nothing other than reset the clock *clk_pls* to schedule the next preemption, T_P time units later. After *MoSkip* has executed, $MONITOR_1$ resumes.

Since the formal discrete to continuous translation schema (24) in general, and its retrenchment correctness PO incarnation (27) in particular, each assume that time progresses in both machines at the same rate, we observe that (27) does not actually connect the two machines unless both *MONITOR* transitions last for exactly T_P time units. We can cater for cases other than this by constructing an $(m, 2n-1)$ diagram, where $m = 1$ (since the *ActContMch*_0.01 machine, after initialisation executes exactly one $MONITOR_0$ transition), and n is the duration of the *ActContMch*_1.01 execution as a multiple of T_P, allowing for $n-1$ skips — or, pushing the formalism a bit, $n = \infty$ if the *ActContMch*_1.01 execution doesn't terminate. In this way, the formalism developed earlier in this paper captures the simulation of an arbitrary *ActContMch*_1.01 execution by *ActContMch*_0.01, albeit that the details are rather trivial.

As a coda to this discussion, we observe that the HEB notion of refinement (slightly different from Section 3, see [12, 13]) *does* permit *ActContMch*_1.01 to be a refinement of *ActContMch*_0.01. The *MoSkip* events of *ActContMch*_1.01 at the source of the problem are viewed as refinements of 'virtual skips' in *ActContMch*_0.01; i.e. of mode events that do not change the state of *ActContMch*_0.01, so can be seen as having taken place or as not having taken place, as convenient. This is a generalisation of the Event-B [3] notion of refinement, which can thus be seen as permitting a certain kind of (m, n) diagram that embodies this virtual skip idea.

13.2 *ActContMch*_1.01 and *ActContMch*_2.5 — First Version

Unlike a normal input device, which, being an engineered object, has a precisely defined set of input values that it can deliver to a digital system, an earthquake is a natural phenomenon, and so any attempt to quantify its behaviour is subject to uncertainty. Despite this, the results of Section 13.1 are mathematically exact. The reason for this is that the earthquake input *e*? enters the models *ActContMch*_0.01 and *ActContMch*_1.01 in 'exactly the same way', and thereby cancels out in considering the relationship between them. However, the earthquake input enters the models *ActContMch*_1.01 and *ActContMch*_2.5 in different ways and these no longer cancel automatically. In particular, in *ActContMch*_1.01, *e*? is the natural physical phenomenon, and so assumptions about it are always contingent to some degree; whereas in *ActContMch*_2.5, the core of the model uses *e*19, which is a sampled value derived indirectly from *e*?. An unavoidable degree of uncertainty thus remains when discussing the relationship between *ActContMch*_1.01 and *ActContMch*_2.5, which we must bear in mind below.

In *ActContMch*_1.01, after initialisation, pliant event $MONITOR_1$ and mode event $MoSkip_1$ interleave indefinitely. In *ActContMch*_2.5, after initialisation, pliant event $MONITOR_2$ is interrupted by the sampling mode events $Sample\_18_2$ and $Sample\_19_2$, after each of which it resumes, being interrupted again by one of the mode events $PulseNo_2$, $PulseMaybe_2$, $PulseYesY_2$, $PulseYesE_2$. After this, the cycle repeats.

We note that *ActContMch*_1.01 has a *p* variable, using which, the $MONITOR_1$ event attempts to compensate for the earthquake signal *e*?; whereas *ActContMch*_2.5 has no such variable, so it attempts to compensate for the earthquake by jolting the building velocity *yy* as needed. There are also no counterparts of the $Sample\_18_2$ and $Sample\_19_2$ events in *ActContMch*_1.01. All of this precludes creating a relationship between *ActContMch*_1.01 and *ActContMch*_2.5 based on

refinements and retrenchments between individual events. Instead, the appropriate thing to do is to consider the cycles of behaviour in the two machines in their entirety, and to create a relationship between *ActContMch*_1.01 and *ActContMch*_2.5 on that basis. This entails using the notions of (m, n) diagram discussed at the end of Section 6. In this section we will therefore consider:

$$\begin{array}{c} MONITOR_1 \fatsemi MoSkip_1 \\ \circledast \\ MONITOR_2 \fatsemi Sample\_18_2 \fatsemi MONITOR_2 \fatsemi Sample\_19_2 \fatsemi MONITOR_2 \fatsemi \mathsf{Last}_2 \end{array} \tag{112}$$

where Last_2 is one of $\{PulseNo_2, PulseMaybe_2, PulseYesY_2, PulseYesE_2\}$, and $\circledast$ is one of $\{\geqslant, \succcurlyeq\}$. We economise on the verbosity of (112) by abbreviating it to $MONskip_1 \circledast MON[\mathsf{last}]_2$ for the purposes of referring to the formal relationship between *ActContMch*_1.01 and *ActContMch*_2.5.

We note that $Sample\_18_2$ and $Sample\_19_2$ merely act as oracles for the $x18$, $x19$ and $e19$ values used by the Last_2 events, so they do not have any impact on the subsequent $MONITOR_2$ behaviour. Also $MoSkip_1$ has no impact on the $MONITOR_1$ behaviour. Accordingly, the essence of the relationship we need to consider can be expressed as $MONITOR_1 \circledast MONITOR_2 \fatsemi \mathsf{Last}_2$.

To address this, consider one iteration of the cycle, which we assume starts at time $\mathbb{t}_\mathrm{L}$ and finishes at time $\mathbb{t}_\mathrm{R}$, where $\mathbb{t}_\mathrm{R} - \mathbb{t}_\mathrm{L} = T_P$. Suppose given (concrete, *ActContMch*_2.5) starting values $xx(\mathbb{t}_\mathrm{L})$ and $yy(\mathbb{t}_\mathrm{L})$, and ending values $xx(\mathbb{t}_\mathrm{R})$ and $yy(\mathbb{t}_\mathrm{R})$, where (in particular) $yy(\mathbb{t}_\mathrm{R})$ is the after-value produced by Last_2 in those cases where Last_2 has a nontrivial effect. We then seek a(n abstract, *ActContMch*_1.01) function $p(t)$ for $t \in [\mathbb{t}_\mathrm{L} \ldots \mathbb{t}_\mathrm{R})$, such that the initial value problem expressed by the SOLVE clause of $MONITOR_1$, with initial values $x(\mathbb{t}_\mathrm{L}) = xx(\mathbb{t}_\mathrm{L})$ and $y(\mathbb{t}_\mathrm{L}) = yy(\mathbb{t}_\mathrm{L})$, delivers final values $x(\mathbb{t}_\mathrm{R}) = xx(\mathbb{t}_\mathrm{R})$ and $y(\mathbb{t}_\mathrm{R}) = yy(\mathbb{t}_\mathrm{R})$ at the end of the $[\mathbb{t}_\mathrm{L} \ldots \mathbb{t}_\mathrm{R})$ interval. This is a typical optimal control problem, discussed extensively in the literature [22, 31, 39, 57, 67, 68], albeit we have refrained from stating any specific optimality criterion yet.

The easiest cases are when Last_2 is one of $PulseNo_2, PulseMaybe_2$. For these cases, there is no contribution from the control system during $MONITOR_2$, and none during $PulseNo_2$ or $PulseMaybe_2$. So setting $p(t)$ to 0 for $t \in [\mathbb{t}_\mathrm{L} \ldots \mathbb{t}_\mathrm{R})$ during $MONITOR_1$ leads to identical behaviour in the two models, which could be captured in a refinement expressed using projection relations, as in Section 13.1. Unfortunately, when we include the nontrivial cases $PulseYesY_2$ and $PulseYesE_2$, the identity of behaviour for the state variables breaks down, and this makes it impossible to assert the preservation of a sensible gluing relation (especially one expressed using projection relations) throughout the $[\mathbb{t}_\mathrm{L} \ldots \mathbb{t}_\mathrm{R})$ interval, as would be required for a refinement.

The remaining cases, $PulseYesY_2$ and $PulseYesE_2$, constitute nontrivial instances of the optimal control problem. The absence of any optimality criterion thus far is connected with the fact that the primary focus of this paper is on structural relationships in system development paths more than on some specific engineering criteria. We take advantage of this now, to absolve ourselves from the need to plunge into the intricacies of Hamilton-Jacobi theory and the Pontryagin Principle.

For a given $[\mathbb{t}_\mathrm{L} \ldots \mathbb{t}_\mathrm{R})$ interval, an *ActContMch*_2.5 execution gives us values $xx(\mathbb{t}_\mathrm{L})$, $yy(\mathbb{t}_\mathrm{L})$, $xx(\mathbb{t}_\mathrm{R})$, $yy(\mathbb{t}_\mathrm{R})$. If we stipulate that these should correspond to the *ActContMch*_1.01 values $x(\mathbb{t}_\mathrm{L})$, $y(\mathbb{t}_\mathrm{L})$, $x(\mathbb{t}_\mathrm{R})$, $y(\mathbb{t}_\mathrm{R})$, we can construct the Hermite interpolating polynomial [65] that interpolates the given values and their derivatives at the two endpoints of the interval. This gives a behaviour for the x and y variables in $MONskip_1$ that matches the xx and yy variables in $MON[\mathsf{last}]_2$ at the endpoints of the interval — but in particular without the impulsive jolt provided by $PulseYesY_2$ or $PulseYesE_2$.

Let $x_\mathrm{H}(t)$ be the Hermite interpolant; its explicit form as a cubic in t is [65]:

$$\begin{aligned} x_{\mathrm{H}}(t) \;\equiv\; & x(\mathbb{t}_{\mathrm{L}})\frac{1}{(\mathbb{t}_{\mathrm{L}}-\mathbb{t}_{\mathrm{R}})^3}(t-\mathbb{t}_{\mathrm{R}})^2[(\mathbb{t}_{\mathrm{L}}-\mathbb{t}_{\mathrm{R}})+2(\mathbb{t}_{\mathrm{L}}-t)]+\\ & y(\mathbb{t}_{\mathrm{L}})\frac{1}{(\mathbb{t}_{\mathrm{L}}-\mathbb{t}_{\mathrm{R}})^2}(t-\mathbb{t}_{\mathrm{L}})(t-\mathbb{t}_{\mathrm{R}})^2+\\ & x(\mathbb{t}_{\mathrm{R}})\frac{1}{(\mathbb{t}_{\mathrm{R}}-\mathbb{t}_{\mathrm{L}})^3}(t-\mathbb{t}_{\mathrm{L}})^2[(\mathbb{t}_{\mathrm{R}}-\mathbb{t}_{\mathrm{L}})+2(\mathbb{t}_{\mathrm{R}}-t)]+\\ & y(\mathbb{t}_{\mathrm{R}})\frac{1}{(\mathbb{t}_{\mathrm{R}}-\mathbb{t}_{\mathrm{L}})^2}(t-\mathbb{t}_{\mathrm{L}})^2(t-\mathbb{t}_{\mathrm{R}}) \end{aligned} \tag{113}$$

It is evident by inspection that (113) satisfies the claimed properties. From $x_{\mathrm{H}}(t)$ we can derive the needed control function $p_{\mathrm{H}}(t)$, as is clear from (111):

$$p_{\mathrm{H}}(t) = [m\,\mathcal{D}^2 + c\,\mathcal{D} + k + m\,e?]\,x_{\mathrm{H}}(t) \tag{114}$$

If we further notice that, being a cubic, $x(t)$ satisfies $\mathcal{D}^4\,x(t) = 0$, then we can add the optimality criterion:

$$\text{Minimise:} \qquad \int_{\mathbb{t}_{\mathrm{L}}}^{\mathbb{t}_{\mathrm{R}}} [\mathcal{D}^4\,x(t)]^2 dt \tag{115}$$

to our *ad hoc* construction, whereby it becomes the solution to an optimal control problem of a normal kind, since it achieves the minimum possible value of (115).

From p_{H} we can get the pp mentioned in $MONITOR_1$ of Fig. 5. Now, assuming the the bound PP_B is adequate, so that all the assumptions made in $MONITOR_1$ can be satisfied, we can construct a formal relationship $MONskip_1 \circledast MON[\mathrm{last}]_2$ as follows. Below, $x19$', $y19$' and $e19$' denote the after-values of $x19$, $y19$ and $e19$ upon their update at time $19\,T_P/20$ during a T_P interval.[19]

- $\circledast$ is $\succcurlyeq$ (116)

  We have already confirmed that a refinement under reasonable conditions is not possible.
- $G_{1,2}(x, y, p; x18, x19, y19, e19, xx, yy) \;\equiv\; x = xx \wedge y = yy$ (117)

  $G_{1,2}$, is an abutment of two projections, thus a regular relation [8, 9]. Recall that $G_{1,2}$ only needs to hold at initialisation.

$\underline{MONskip_1 \succcurlyeq MON[PulseNo]_2}$

- $W_{MONskip_1 \succcurlyeq MON[PulseNo]_2}(\ldots) \;\equiv\; x(\mathbb{t}_{\mathrm{L}}) = xx(\mathbb{t}_{\mathrm{L}}) \wedge y(\mathbb{t}_{\mathrm{L}}) = yy(\mathbb{t}_{\mathrm{L}}) \wedge |x19'| < X_{th} < X_B \wedge$ **"**$e/e?$-Eq**"** (118)
- $D_{MONskip_1 \succcurlyeq MON[PulseNo]_2}(\ldots) \;\equiv\; x(\mathbb{t}_{\mathrm{R}}) = xx(\mathbb{t}_{\mathrm{R}}) \wedge y(\mathbb{t}_{\mathrm{R}}) = yy(\mathbb{t}_{\mathrm{R}}) \wedge |xx(\mathbb{t}_{\mathrm{R}})| \leq X_B \wedge$ **"**No-Xtra**"** (119)

"$e/e?$-Eq**"** denotes $|\,e?_1(t)\,| \leq E_B \wedge |\,e?_2(t)\,| \leq E_B \wedge e_1 = e_2 \wedge e?_1(t) = e?_2(t)$.[20] In **"**$No$-Xtra**"** can be placed additional assertions regarding the final (and perhaps other) values of xx and yy that strengthen the bound $|xx(\mathbb{t}_{\mathrm{R}})| \leq X_B$.

[19] Occurrences of variable values at the extremes of the sampling period, $\mathbb{t}_{\mathrm{L}}$ and $\mathbb{t}_{\mathrm{R}}$ are to be interpreted as the relevant limiting values where necessary.

[20] Unlike HEB, whose refinement semantics automatically assumes that identically named abstract/concrete variables are equal, the formalism of Sections 3 and 4 does not do so explicitly. Hence the presence of **"**$e/e?$-Eq**"**.

$\underline{MONskip_1 \succcurlyeq MON[PulseMaybe]_2}$

- $W_{MONskip_1 \succcurlyeq MON[PulseMaybe]_2}(\ldots) \equiv x(\mathbb{t}_L) = xx(\mathbb{t}_L) \wedge y(\mathbb{t}_L) = yy(\mathbb{t}_L) \wedge$
  $X_{th} \leq |x19'| \leq X_B \wedge$ “$e/e?$-Eq” $\wedge$ “P-Dsc’-2” $\leq X_B$ (120)
- $D_{MONskip_1 \succcurlyeq MON[PulseMaybe]_2}(\ldots) \equiv x(\mathbb{t}_R) = xx(\mathbb{t}_R) \wedge y(\mathbb{t}_R) = yy(\mathbb{t}_R) \wedge |xx(\mathbb{t}_R)| \leq X_B \wedge$
  “*Maybe*-Xtra” (121)

  “$e/e?$-Eq” is as in (118). “P-Dsc’-2” denotes the expression $\left| x19' \left(1 - \omega^2 T_P^2/2\right) + y19' T_P \left(1 - \zeta\, \omega\, T_P\right) - e19' T_P^2/2 \right|$. Its value compared with X_B discriminates cases where a pulse is generated or not in *ActContMch_*2.5. In “*Maybe*-Xtra” can be placed additional assertions regarding the final (and perhaps other) values of xx and yy that strengthen the bound $|xx(\mathbb{t}_R)| \leq X_B$.

$\underline{MONskip_1 \succcurlyeq MON[PulseYesY]_2}$

- $W_{MONskip_1 \succcurlyeq MON[PulseYesY]_2}(\ldots) \equiv x(\mathbb{t}_L) = xx(\mathbb{t}_L) \wedge y(\mathbb{t}_L) = yy(\mathbb{t}_L) \wedge$
  $X_{th} \leq |x19'| \leq X_B \wedge$ “$e/e?$-Eq” $\wedge$ “P-Dsc’-2” $> X_B \wedge$ “Y.gtr.E” (122)
- $D_{MONskip_1 \succcurlyeq MON[PulseYesY]_2}(\ldots) \equiv x(\mathbb{t}_R) = xx(\mathbb{t}_R) \wedge y(\mathbb{t}_R) = yy(\mathbb{t}_R) \wedge |xx(\mathbb{t}_R)| \leq X_B \wedge$
  “*YesY*-Xtra” (123)

  “$e/e?$-Eq” is as in (118). “P-Dsc’-2” is as in (120). “Y.gtr.E” indicates the additional conditions in the guard of *PulseYesY* in Fig. 7 that assert that the yy contribution is greater than the $e19$ contribution in potentially breaching the X_B bound if action is not taken. In “*YesY*-Xtra” can be placed additional assertions regarding the final (and perhaps other) values of xx and yy that strengthen the bound $|xx(\mathbb{t}_R)| \leq X_B$.

$\underline{MONskip_1 \succcurlyeq MON[PulseYesE]_2}$

- $W_{MONskip_1 \succcurlyeq MON[PulseYesE]_2}(\ldots) \equiv x(\mathbb{t}_L) = xx(\mathbb{t}_L) \wedge y(\mathbb{t}_L) = yy(\mathbb{t}_L) \wedge$
  $X_{th} \leq |x19'| \leq X_B \wedge$ “$e/e?$-Eq” $\wedge$ “P-Dsc’-2” $> X_B \wedge$ “E.gtr.Y” (124)
- $D_{MONskip_1 \succcurlyeq MON[PulseYesE]_2}(\ldots) \equiv x(\mathbb{t}_R) = xx(\mathbb{t}_R) \wedge y(\mathbb{t}_R) = yy(\mathbb{t}_R) \wedge |xx(\mathbb{t}_R)| \leq X_B \wedge$
  “*YesE*-Xtra” (125)

  “$e/e?$-Eq” is as in (118). “P-Dsc’-2” is as in (120). “E.gtr.Y” indicates the additional conditions in the guard of *PulseYesE* in Fig. 7 that assert that the $e19$ contribution is greater than the yy contribution in potentially breaching the X_B bound if action is not taken. In “*YesE*-Xtra” can be placed additional assertions regarding the final (and perhaps other) values of xx and yy that strengthen the bound $|xx(\mathbb{t}_R)| \leq X_B$.

In the preceding, note that although “$e/e?$-Eq” is explicitly defined, and “Y.gtr.E” and “E.gtr.Y” are just names of complicated expressions in Fig. 7, the various “__-Xtra” clauses indicate further facts that do not appear anywhere, but that could be obtained by combining the more detailed technical discussion in [11] with the optimal control perspective discussed above. This possibility arises because the techniques involved in deriving such properties are highly generic and extendable, and present a lot of opportunities for deriving additional facts.

The construction given above gives an (m, n) diagram, in the sense of Section 6. If we assert that at no moment in the interior of a T_P interval is $G_{1,2}$ established, we have a strong (m, n) diagram.[21] We conclude therefore, that in the terminology of Section 6, *ActContMch_*2.5 is refining simulable by *ActContMch_*1.01, using the constructed (m, n) diagrams instead of individual steps.

[21] We can expect this to be the case except, perhaps, for a set of configurations of measure zero in the parameter space.

As a coda to this discussion, we observe that the reestablishing of $G_{1,2}$ at the end of each T_P interval (and disregarding everything else) enables the $MONskip_1 \circledast MON[\text{last}]_2$ (m,n) diagrams we have constructed to be viewed, not only as the (m,n) diagrams that were discussed in Section 6, but as as the (m,n) diagrams of a *bona fide* ASM refinement [27]. Moreover, the fact that we can simulate *any* *ActContMch_*2.5 execution (whose duration is an integral number of T_P intervals) using a series of such diagrams, means that in the terminology of Section 6, the construction of this section makes *ActContMch_*2.5 strongly comprehensively simulable via this collection of (m,n) diagrams.

As a further coda to the discussion, we point out that although the simulation via the (m,n) diagrams just discussed works perfectly well, it disregards the nontrivial invariants in the two models, which require that $|xx(t)| \leq X_B$ and $|x(t)| \leq X_B$ hold at all times. We should pay some attention to these.

For xx, the arguments in [11] that justify the details of the *ActContMch_*2.5 machine (e.g. the complicated tests that select between the various [last] cases), are aimed precisely at ensuring that $|xx| \leq X_B$ holds at all times, so this invariant needs no further discussion.

For x, we can make use of the preceding in the following way. The analysis in [11], based on the constants and bounds appearing in the *ActContMch_*2.5 machine, is aimed at ensuring that for any T_P interval, even if the velocity increases at the fastest rate permitted by those constants and bounds, $xx(\mathbb{t}_L)$ and $yy(\mathbb{t}_L)$ remain small enough that $|xx| \leq X_B$ cannot be breached within that interval. (The jolt to yy (if any) that is applied at $\mathbb{t}_R$ is aimed at ensuring the same for the *next* T_P interval, given that no help will be given by the control system till the end of the next interval.) So, we can assume that neither $|xx(\mathbb{t}_L)|$ nor $|xx(\mathbb{t}_R)|$ exceed X_B, and thus $-B_X \leq \mathsf{MIN} \equiv \min\{xx(\mathbb{t}_L), xx(\mathbb{t}_R)\} \leq \max\{xx(\mathbb{t}_L), xx(\mathbb{t}_R)\} \equiv \mathsf{MAX} \leq B_X$.

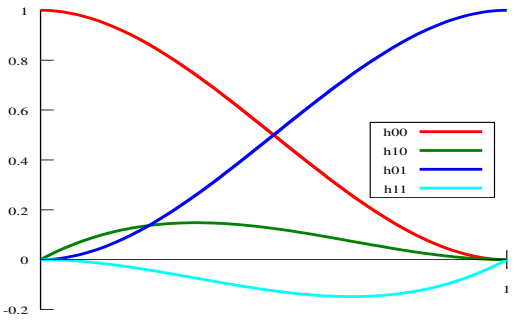

Fig. 8. The Hermite interpolant basis functions, normalised to the unit interval. From [77].

Now, we know that $x(\mathbb{t}_L) = xx(\mathbb{t}_L)$ and $x(\mathbb{t}_R) = xx(\mathbb{t}_R)$ by construction, and that the trajectory of x from $\mathbb{t}_L$ to $\mathbb{t}_R$ is a linear combination of Hermite basis functions, illustrated in Fig. 8, with $h00$, $h10$, $h01$, $h11$ corresponding to the $x(\mathbb{t}_L)$, $y(\mathbb{t}_L)$, $x(\mathbb{t}_R)$, $y(\mathbb{t}_R)$ terms in (113) respectively.

For $t \in [\mathbb{t}_L \ldots \mathbb{t}_R]$, it can be seen that $h00(t) + h01(t) = 1$, so that the first and third terms of (113) form a convex sum of $x(\mathbb{t}_L)$ and $x(\mathbb{t}_R)$, which therefore remains within the range $[\mathsf{MIN} \ldots \mathsf{MAX}]$.

For the second and fourth terms of (113), the worst case is when $y(\mathbb{t}_L)$ is maximal towards a B_X boundary and $y(\mathbb{t}_R)$ is maximal away from the same B_X boundary. In such a case, the maximal absolute value of velocity $y_{\max}$ can be deduced from [11], and the two terms amount to a displacement:

$$disp_{\max} = y_{\max}\,(t - \mathbb{t}_L)(\mathbb{t}_R - t)/T_P \tag{126}$$

which reaches its maximum value half way between $\mathbb{t}_L$ and $\mathbb{t}_R$. The threshold value X_{th} can therefore be reduced to accommodate any potential overshoot that arises for this reason. Whether it is desirable to do so, is however, debatable. The behaviour in (113) satisfies an existential criterion; so there is no reason why a different existential witness could not do better. Given the critique of this section that appears in the next version, we do not pursue the details further.

13.3 *ActContMch*_1.01 and *ActContMch*_2.5 — Second Version

In the preceding section we found that, within the limits prescribed by the various parameters and constants, for every $x18$, $x19$ pair, we could derive a behaviour of the earthquake and of p/pp that conformed exactly to the implications that follow from the $x18$, $x19$ values. This is a consequence of the orientation of the implication in relationships like the refinement correctness PO (6) and the retrenchment correctness PO (19). But from a real world perspective, this is a strange thing to do — earthquakes hardly ask permission from a sampled system model regarding what their their behaviour should be over the next T_P interval, the more so considering that the sampled model will give answers that are always approximations to any related continuous behaviour that they are derived from.[22] Even less is it the case that an earthquake would feel obliged to follow exactly the behaviour x_{H} discussed above.

A better question is thus whether: for any earthquake behaviour $e?$ (within the assumed limits), and for appropriate p/pp control which ensures, in *ActContMch*_1.01, that the whole system stays within safe limits in a given T_P interval (taking into account all the constraints that are assumed to hold), there is a $x18$, $x19$ pair (based on the same $e?$ but without any p/pp intervention), such that the impulse based control of *ActContMch*_2.5 produces system behaviour that agrees in a suitable way with the system behaviour in *ActContMch*_1.01. This inverts the implication in relationships like (6) and (19). In this section we will therefore consider:

$$\begin{array}{c} MONITOR_2 \fatsemi Sample\_18_2 \fatsemi MONITOR_2 \fatsemi Sample\_19_2 \fatsemi MONITOR_2 \fatsemi \mathsf{Last}_2 \\ \circledast \\ MONITOR_1 \fatsemi MoSkip_1 \end{array} \tag{127}$$

or, more briefly, $MON[\mathsf{last}]_2 \circledast MONskip_1$, inverting (112).

The derivation in Section 7 of [11], addresses essentially this question, but with the view that going from *ActContMch*_1 to *ActContMch*_2 is a 'refinement' — without taking care to confirm whether the technical details investigated there align with the implications of a formal refinement relationship of any particular kind, as we discuss here. We summarise the argument of Section 7 of [11]; see *loc. cit.* for full details.

The active control problem of *ActContMch*_1.01 is a linear ODE system with constant coefficients, so has a solution in closed form [4, 29, 72, 76], which we can succinctly write as:

$$[x(t), y(t)]^{\mathrm{T}} = \mathrm{e}^{\boldsymbol{A}(t-\mathbb{t}_{\mathrm{L}})}[x(\mathbb{t}_{\mathrm{L}}), y(\mathbb{t}_{\mathrm{L}})]^{\mathrm{T}} + (\mathrm{e}^{\boldsymbol{A}\,s} *_{s\in[\mathbb{t}_{\mathrm{L}}\ldots t]} [0, pp(s)/m - e?(s)]^{\mathrm{T}}) \tag{128}$$

and which appears, abbreviated, in Fig. 5. When the details of the companion matrix $\boldsymbol{A}$ are disentangled, (128) yields a Duhamel integral with specified initial values for the dynamics. See [30, 78]. Given the assumed bounds on earthquake behaviour, the COMPLY clause of *MONITOR* in Fig. 5 guarantees that the solution $[x(t), y(t)]^{\mathrm{T}}$ in any T_P interval satisfies the invariant $|x(t)| \leq X_B$.

As discussed in [11], the appropriate damping factor of the system has value about $\zeta \lesssim 0.1$, and the frequency of the active control system is chosen to be about 20 times the natural frequency of the earthquake signal, so that $\omega t \leq \omega T_P \lesssim 0.05$. This leads to some simplification. Specifically, the natural and damped frequencies of the system differ only negligibly, and both are called ω below. The Duhamel integral then becomes:

[22] Note that a different argument could be raised if the key driver in the system behaviour was not an external influence like an earthquake, but a human-determined control. Then, asking whether for every discrete control there was a corresponding continuous control, would potentially make more sense.

$$
\begin{aligned}
x(t-\mathbb{t}_{\mathrm{L}}) \;=\; & \\
& \mathrm{e}^{-\zeta\,\omega\,(t-\mathbb{t}_{\mathrm{L}})}\left[x(\mathbb{t}_{\mathrm{L}})\cos(\omega(t-\mathbb{t}_{\mathrm{L}}))+\omega^{-1}\left(y(\mathbb{t}_{\mathrm{L}})+\zeta\,\omega\,x(\mathbb{t}_{\mathrm{L}})\right)\sin(\omega(t-\mathbb{t}_{\mathrm{L}}))\right] \\
& +\frac{1}{\omega}\int_{0}^{(t-\mathbb{t}_{\mathrm{L}})} a_X(s)\,\mathrm{e}^{-\zeta\,\omega((t-\mathbb{t}_{\mathrm{L}})-s)}\sin(\omega((t-\mathbb{t}_{\mathrm{L}})-s))\,ds
\end{aligned}
\tag{129}
$$

$$
\begin{aligned}
y(t-\mathbb{t}_{\mathrm{L}}) \;=\; \mathcal{D}\,x(t-\mathbb{t}_{\mathrm{L}}) \;=\; & \\
& \mathrm{e}^{-\zeta\,\omega(t-\mathbb{t}_{\mathrm{L}})}\left[y(\mathbb{t}_{\mathrm{L}})\cos(\omega(t-\mathbb{t}_{\mathrm{L}}))-\left(\zeta\,y(\mathbb{t}_{\mathrm{L}})+\omega\,x(\mathbb{t}_{\mathrm{L}})\right)\sin(\omega(t-\mathbb{t}_{\mathrm{L}}))\right] \\
& +\int_{0}^{(t-\mathbb{t}_{\mathrm{L}})} a_X(s)\,\mathrm{e}^{-\zeta\,\omega((t-\mathbb{t}_{\mathrm{L}})-s)}\left(\cos(\omega((t-\mathbb{t}_{\mathrm{L}})-s))-\zeta\sin(\omega((t-\mathbb{t}_{\mathrm{L}})-s))\right)ds
\end{aligned}
\tag{130}
$$

where $a_X(s)$ is the externally imposed acceleration $a_X(s) = (pp(s)/m - e?(s))$.

To properly relate any specific instance of an *ActContMch*_1.01 behaviour to the corresponding *ActContMch*_2.5 behaviour that depends on the correct $x18$, $x19$ and $e19$ values, we would have to do the following. For all admissible $e?$ behaviours and pp controls that ensured that the $|x(t)| \leq X_B$ invariant was always respected, we would have to extract an $e19$ value from the $e?$ function at time $19\,T_P/20$, and would have to extract $x18$ and $x19$ values at times $18\,T_P/20$ and $19\,T_P/20$ respectively from a solution of (129) depending on the same $e?$ but with pp set to zero throughout. This is a tall order, given the fact that the quantifications on $e?$ and pp are given so indirectly. Instead, we can proceed as we did in [11].

Since T_P is one twentieth of a natural period of the sinusoids in (129)-(130), those sinusoids have the same sign throughout the range of integration. This allows further simplification and estimation of the integrals, through the substitution of various terms by their maximal values, and evaluation to a closed form.

Retaining only leading terms of the sinusoids in the result, leads to the formulation used in the creation of the model in Fig. 6. In matrix form, for an interval of duration $\tau \leq T_P$ starting from initial values at $\mathbb{t}_{\mathrm{L}}$, for a constant externally imposed acceleration a_X, and assuming that the earthquake is parameterised by constants e and $e?$, with constant compensating force pp, so that $a_X = pp/m - (e \pm e_B)$, with the sign of e_B chosen to maximise the end of period displacement, the closed form is:

$$
\begin{bmatrix} xx(\tau+\mathbb{t}_{\mathrm{L}}) \\ yy(\tau+\mathbb{t}_{\mathrm{L}}) \end{bmatrix} = \begin{bmatrix} \left(1-\omega^2\,\tau^2/2\right) & \tau\left(1-\zeta\,\omega\,\tau\right) \\ -\omega^2\,\tau & \left(1-2\,\zeta\,\omega\,\tau\right) \end{bmatrix} \begin{bmatrix} xx(\mathbb{t}_{\mathrm{L}}) \\ yy(\mathbb{t}_{\mathrm{L}}) \end{bmatrix} - \begin{bmatrix} a_X\,\tau^2/2 \\ a_X\,\tau\left(1-\zeta\,\omega\,\tau\right) \end{bmatrix}
\tag{131}
$$

As mentioned in the previous section, when based on the most pessimistic values for $e \pm e_B$, and on $pp(t) = 0$, (131) allows the calculation of a threshold value X_{th} around which machine *ActContMch*_2.5 can determine whether the $pp(t) = 0$ policy of machine *ActContMch*_2.5 will cause the $|xx(t)| \leq X_B$ invariant to be respected in the next T_P interval or not. This delegates the determination of the correct $e19$, $x18$ and $x19$ values, and more importantly, the subsequent *ActContMch*_2.5 behaviour that is defined using those values, to the scheduling of events in *ActContMch*_2.5 'at runtime'.

Thus the mechanisms through which machines *ActContMch*_1.01 and *ActContMch*_2.5 assure their invariants are very different. This makes the eliciting of the details of a formal relationship such as $MON[\mathrm{last}]_2 \circledast MONskip_1$ more challenging. Regarding this, we make the following comments.

Unlike in the first version of the *ActContMch*_1.01 and *ActContMch*_2.5 relationship, we are not in a position to 'compel' one of the systems to behave like the other one, so we cannot ask that x and xx, and y and yy are the same at the two ends of a T_P interval. Therefore, defining G to be as in the first version, will not lead to a provable strongly comprehensively simulable relationship.

Therefore, the best we can reasonably do is to define G to be the conjunction of the $|x/xx| \leq X_B$ conditions in the two models. Beyond that, we observe that the various conditions that figured in the $MONskip_1 \circledast MON[\text{last}]_2$ relationship earlier did not depend on the orientation of the implication in (19). This permits us to reuse their structure in the present case, provided we remove any equalities between state variables, which are inappropriate here — we must ensure that such conditions are also not present in the various "XX-Xtra" clauses introduced earlier, which we denote using a bullet decoration. Also, $x19'$, $y19'$, $e19'$, "$e/e?$-Eq" and "P-Dsc'-2" are as before. In this manner we derive a formal $MON[\text{last}]_2 \circledast MONskip_1$ relationship as follows.

- $\circledast$ is $\succcurlyeq$ (132)
- $G_{2,1}(x, y, p; x18, x19, y19, e19, xx, yy) \equiv |x| \leq X_B \wedge |xx| \leq X_B$ (133)

$\underline{MON[PulseNo]_2 \succcurlyeq MONskip_1}$

- $W_{MON[PulseNo]_2 \succcurlyeq MONskip_1}(\ldots) \equiv |xx(\mathbb{t}_L)| \leq X_B \wedge |x(\mathbb{t}_L)| \leq X_B \wedge$ "$e/e?$-Eq" $\wedge$ $|x19'| < X_{th} < X_B$ (134)
- $D_{MON[PulseNo]_2 \succcurlyeq MONskip_1}(\ldots) \equiv |x(\mathbb{t}_R)| \leq X_B \wedge |xx(\mathbb{t}_R)| \leq X_B \wedge$ "No-Xtra$^\bullet$"

$\underline{MON[PulseMaybe]_2 \succcurlyeq MONskip_1}$

- $W_{MON[PulseMaybe]_2 \succcurlyeq MONskip_1}(\ldots) \equiv |xx(\mathbb{t}_L)| \leq X_B \wedge |x(\mathbb{t}_L)| \leq X_B \wedge$ "$e/e?$-Eq" $\wedge$ $X_{th} < |x19'| \leq X_B \wedge$ "P-Dsc'-2" $\leq X_B$ (135)
- $D_{MON[PulseMaybe]_2 \succcurlyeq MONskip_1}(\ldots) \equiv |x(\mathbb{t}_R)| \leq X_B \wedge |xx(\mathbb{t}_R)| \leq X_B \wedge$ "$Maybe$-Xtra$^\bullet$" (136)

$\underline{MON[PulseYesY]_2 \succcurlyeq MONskip_1}$

- $W_{MON[PulseYesY]_2 \succcurlyeq MONskip_1}(\ldots) \equiv |xx(\mathbb{t}_L)| \leq X_B \wedge |x(\mathbb{t}_L)| \leq X_B \wedge$ "$e/e?$-Eq" $\wedge$ $X_{th} < |x19'| \leq X_B \wedge$ "P-Dsc'-2" $> X_B \wedge$ "Y.gtr.E" (137)
- $D_{MON[PulseYesY]_2 \succcurlyeq MONskip_1}(\ldots) \equiv |x(\mathbb{t}_R)| \leq X_B \wedge |xx(\mathbb{t}_R)| \leq X_B \wedge$ "$YesY$-Xtra$^\bullet$" (138)

$\underline{MON[PulseYesE]_2 \succcurlyeq MONskip_1}$

- $W_{MON[PulseYesE]_2 \succcurlyeq MONskip_1}(\ldots) \equiv |xx(\mathbb{t}_L)| \leq X_B \wedge |x(\mathbb{t}_L)| \leq X_B \wedge$ "$e/e?$-Eq" $\wedge$ $X_{th} < |x19'| \leq X_B \wedge$ "P-Dsc'-2" $> X_B \wedge$ "E.gtr.Y" (139)
- $D_{MON[PulseYesE]_2 \succcurlyeq MONskip_1}(\ldots) \equiv |x(\mathbb{t}_R)| \leq X_B \wedge |xx(\mathbb{t}_R)| \leq X_B \wedge$ "$YesE$-Xtra$^\bullet$" (140)

As in the previous version, given the different G, which does not demand the truth of something we cannot prove, the construction above gives an (m, n) diagram, in the sense of Section 6. As before, we can therefore conclude that *ActContMch*_1.01 is refining simulable by *ActContMch*_2.5, using the constructed (m, n) diagrams.

We observe, finally, that the refining simulability just presented is actually *Init*-constrained, with $G_{1,2}$ from the previous version as the *Init*-constraint. This holds at initialisation since the states of both *ActContMch*_1.01 and *ActContMch*_2.5 are initialised at zero. Had the *ActContMch*_2.5 initialisation been allowed to deviate too far from zero, it is not guaranteed that the dynamics (with zero pp) would remain safe, since the safety during any T_P interval relies on measures taken during the *previous* T_P interval.

13.4 *ActContMch*_1.01 and *ActContMch*_2.5 — Third Version

In this third version of the relationship between *ActContMch*_1.01 and *ActContMch*_2.5, we focus on the simulation properties explored in Sections 7 and 8. These all rely on a common infrastructure, which we address first.

13.4.1 **Common Infrastructure.** Firstly, the relevant parts of the state spaces of machines *ActContMch*_1.01 and *ActContMch*_2.5 are both subsets of $\mathbb{R}\times\mathbb{R}$, connected by an identity isomorphism. Similarly for other variables. This means that the natural distances between variable values, as needed for Section 8.1, can be computed by referring to their values in $\mathbb{R}$, without regard for whether they are abstract or concrete variables.

Furthermore, many of the results in Section 8.1 assume specific values for $\Delta_G, \Delta_I, \Delta_O$, and whereas different values of Δ_G are justifiable in different circumstances, the fact that both *ActContMch*_1.01 and *ActContMch*_2.5 consume the same $e?$ suggests that it is sufficient to always have $\Delta_I = 0$ in the no-origin versions of the Section 8 metric results. Moreover, the absence of outputs in our models allows us to elide all occurrences of output relations and of any Δ_O they may rely on when instantiating generic results.

We argued in the previous versions, that the analysis in [11] identified a region of $(xx(\mathbb{t}_{\mathrm{L}}), yy(\mathbb{t}_{\mathrm{L}}))$ values, within which, provided at least one T_P interval has previously elapsed (to apply impulsive control if it is required), an *ActContMch*_2.5 trajectory was guaranteed to safely remain, and that the same region would therefore also do for *ActContMch*_1.01. We also know from the previous versions, that the dynamics in both models is confined to the region $|x/xx| \leq X_B$, with each such $|x/xx|$ defining a range of $|y/yy|$ values that ensures no escape from $|x/xx| \leq X_B$. Let us call the safe region thus defined $\mathrm{Saf}[X_B]$, in both models. The fact that $\mathrm{Saf}[X_B]$ is not the same as the $|x/xx| \leq X_B$ invariant region, highlights the fact that the reachable subspace in either model is a proper subset of that invariant.

Secondly, the simulation results of Sections 7 and 8 are asymmetric between abstract and concrete systems, so we must decide how this maps to *ActContMch*_1.01 and *ActContMch*_2.5. In the present, third version, we use the perspective of the second version, with *ActContMch*_2.5 as abstract model. An alternative could be patterned in a similar way to the first version.

Thirdly, a further element needed in various places is the contracting nature of the underlying dynamics. Let us rewrite (131) in terms of dimensionally compatible quantities, $\widetilde{xx} \equiv xx$ and $\widetilde{yy} \equiv yy/\omega$.[23] Then, the matrix in (131) becomes the dimensionless:

$$\boldsymbol{A} = \begin{bmatrix} \left(1-\omega^2\,\tau^2/2\right) & \omega\,\tau\,\left(1-\zeta\,\omega\,\tau\right) \\ -\omega\,\tau & \left(1-2\,\zeta\,\omega\,\tau\right) \end{bmatrix} \tag{141}$$

With the help of *Mathematica* [60], or otherwise, it is easy to find that the eigenvalues of $\boldsymbol{A}$ are:

$$\lambda^{\pm}_{\boldsymbol{A}} = 1-\zeta\,\omega\,\tau-\omega^2\,\tau^2/4 \pm \mathrm{i}\,\omega\,\tau\,\sqrt{1-\zeta\,\omega\,\tau/2-\omega^2\,\tau^2/16-\zeta^2} \tag{142}$$

From this, for small ζ, ω, τ, it is relatively clear that:

$$|\lambda^{\pm}_{\boldsymbol{A}}|^2 = 1-2\,\zeta\,\omega\,\tau+\omega^2\,\tau^2/2+\mathrm{O}(\omega^3\,\tau^3) \tag{143}$$

so that, given the numerical values quoted earlier:

$$|\lambda^{\pm}_{\boldsymbol{A}}| = 1-\zeta\,\omega\,\tau+\mathrm{O}(\omega^2\,\tau^2) < 1 \tag{144}$$

Thus, both eigenvalues have the same magnitude and so shrink their eigenvectors towards the origin by the same amount, which shows that $\boldsymbol{A}$ is contracting (with the state space origin as fixpoint). And since $\boldsymbol{A}$ specifies the leading order contribution of the homogeneous part of the

[23] $\widetilde{xx}$ and $\widetilde{yy}$ are the standard dimensionally compatible quantities used in earthquake engineering.

continuous behaviour in (129)-(130) too, we conclude that the homogeneous part of the dynamics of (129)-(130), which we refer to as $\mathcal{A}$, is also contracting (with the same fixpoint).

We claim that the disjoint union of $\mathcal{A} \oplus \boldsymbol{A}$ is R-adapted, and thus contracting, where R is the natural identity on $\mathbb{R} \times \mathbb{R}$ as state space for variable pairs (x, y) and (xx, yy) respectively. For this we must establish the R-adapted criteria (63)-(64) for the stated R.

But noting that R is an identity, $\mathcal{A}$ and $\boldsymbol{A}$ are both deterministic, and $\mathcal{A} - \boldsymbol{A}$ (when acting on the same vector space) consists of terms which we regard as negligible, this is relatively easy since $\mathcal{A} =^{\mathrm{o}(\omega\tau)} \boldsymbol{A}$. Thus, suppose given a T_P interval, and $(x(\mathbb{t}_\mathrm{L}), y(\mathbb{t}_\mathrm{L}))$ and $(xx(\mathbb{t}_\mathrm{L}), yy(\mathbb{t}_\mathrm{L}))$. If $(xx(\mathbb{t}_\mathrm{L}), yy(\mathbb{t}_\mathrm{L}))$ is R-related to $(\widehat{x}(\mathbb{t}_\mathrm{L}), \widehat{y}(\mathbb{t}_\mathrm{L}))$ in the (x, y) state space, and $(x(\mathbb{t}_\mathrm{L}), y(\mathbb{t}_\mathrm{L}))$ is R-related to $(\widehat{xx}(\mathbb{t}_\mathrm{L}), \widehat{yy}(\mathbb{t}_\mathrm{L}))$ in the (xx, yy) state space, and we apply $\mathcal{A}$ for a duration T_P to $(x(\mathbb{t}_\mathrm{L}), y(\mathbb{t}_\mathrm{L}))$ and $(\widehat{x}(\mathbb{t}_\mathrm{L}), \widehat{y}(\mathbb{t}_\mathrm{L}))$, and we apply $\boldsymbol{A}$ for a for a duration T_P to $(xx(\mathbb{t}_\mathrm{L}), yy(\mathbb{t}_\mathrm{L}))$ and $(\widehat{xx}(\mathbb{t}_\mathrm{L}), \widehat{yy}(\mathbb{t}_\mathrm{L}))$, the shrinkage of the distance between the former pair is matched by the shrinkage of the distance between the latter pair up to negligible terms, and this leads to the metric properties required. The contracting nature of $\mathcal{A} \oplus \boldsymbol{A}$ opens the door to the simulation results of Section 8.

Fourthly, since in this version, we want to connect our metric approach to the results we know from the application domain, which, in particular, focus exclusively on the magnitude of the x component of the state space, it is convenient to use the metric on the $\mathbb{R} \times \mathbb{R}$ state spaces given by:

$$d^X((x_1, y_1), (x_2, y_2)) \;\equiv\; |x_1 - x_2| \tag{145}$$

(so we definitely need the latitude of the 'pseudo-' spoken of in footnote 12).

13.4.2 **Theorem 8.6.** We start with Theorem 8.6, in the no-origin version. This relies on a number of conditions, the first of which is:

$$Init_X(u_X) \wedge Init_Y(u_Y) \Rightarrow G^{\Delta_G}_{X,Y}(u_X, u_Y) \wedge EqEnbl_{X,Y}(u_X, u_Y) \tag{146}$$

Since both abstract and concrete models start with the state at $(0, 0)$, $\Delta_G = 0$ could help satisfy (146). But that would be a bad choice.[24] Instead, since we know, independently, that $|x/xx| < X_B$ always holds, choosing:

$$\Delta_G \;\equiv\; 2X_B \tag{147}$$

will make G^{Δ_G} true for all cases of safe dynamics. In addition, since for all cases of the (m, n) diagrams we considered earlier, both abstract and concrete execution fragments start with *MONITOR* pliant events, i.e. ODEs which have a solution from *any* initial point, the $EqEnbl_{X,Y}$ condition in (146) is trivially satisfied.

The other condition of interest is:

$$\begin{aligned} &G^{\Delta_G}_{X,Y}(u_X, u_Y) \wedge In^{\Delta_I}_{Op_{X,Y}}(i_X, i_Y) \wedge stp_{Op_X}(u_X, i_X, u'_X, o_X) \wedge stp_{Op_Y}(u_Y, i_Y, u'_Y, o_Y) \Rightarrow \\ &\quad (\exists\, \tilde{u}'_X \in U_X, \tilde{o}_X \in O_{Op_X} \bullet stp_{Op_X}(u_X, i_X, \tilde{u}'_X, \tilde{o}_X) \wedge \\ &\qquad G^{\Delta_G}_{X,Y}(\tilde{u}'_X, u'_Y) \wedge Out^{\Delta_O}_{Op_{X,Y}}(\tilde{o}_X, o_Y) \wedge EqEnbl_{X,Y}(\tilde{u}'_X, u'_Y)) \end{aligned} \tag{148}$$

for each (m, n) diagram $stp_{Op_X} \equiv MON[\mathrm{last}]_2 \circledast MONskip_1 \equiv stp_{Op_Y}$. Here, at the end of a partial simulation, we know that $G^{\Delta_G}_{X,Y}$ holds, that $In^{\Delta_I}_{Op_{X,Y}}$ holds, and from $EqEnbl_{X,Y}$, that either both execution fragments or neither will extend beyond (u_X, u_Y).

If it is neither, we are done. Otherwise, since we are following the second version here, the behaviour of the abstract system is entirely determined by that of the concrete system, and the dynamics of both systems is deterministic (once a choice of pp has been made for the concrete

[24] If we were following the first version rather than the second, $\Delta_G = 0$ would be a good choice provided we cared only about the states at the beginning and end of an (m, n) diagram.

system (for a given $e?$ in a given T_P interval)). This enables us to identify $\tilde{u}'_X, \tilde{o}_X$ with u'_X, o_X in (148).[25] For u'_X, u'_Y, we know that $G^{\Delta_G}_{X,Y}$ is true. Also, we said already that $Out_{Op_{X,Y}}$ is irrelevant, and that $EqEnbl_{X,Y}$ is trivially satisfied. So we can use Theorem 8.6 to infer a trace inclusion between the *ActContMch*_2.5 (abstract) and *ActContMch*_1.01 (concrete) systems, based on the W/D data considered for the second version.

If we now go to the with-origin version of the theorem, where we obviously identify the origins with the origins of suitable Cartesian products of $\mathbb{R}$ for the various state and input spaces, the possibilities are parameterised by the triples $(\eta, \alpha_X, \alpha_Y)$ for the various metric spaces. Regarding the state space, knowing already that $|x| < X_B$ and $|xx| < X_B$ both hold, implies that any positive linear combination based on those and on the earlier $\Delta_G = 2X_B$ will yield a $G^{\Delta_G}_{X,Y}$ that enables (148) to be proved for all cases of safe dynamics, provided $In^{\Delta_I}_{Op_{X,Y}}$ is also a positive linear combination. So we can apply Theorem 8.6 to infer a trace inclusion as before. We do not pursue the details further.

13.4.3 **Theorem 8.12.** Next is Theorem 8.12, starting again with the no-origin case. This is the same as Theorem 8.6 aside from the replacement of (148) by (149). Even then, the only difference arises in the third line:

$$\begin{aligned} &G^{\Delta_G}_{X,Y}(u_X, u_Y) \wedge In^{\Delta_I}_{Op_{X,Y}}(i_X, i_Y) \wedge stp_{Op_X}(u_X, i_X, u'_X, o_X) \wedge stp_{Op_Y}(u_Y, i_Y, u'_Y, o_Y) \Rightarrow \\ &\quad (\exists\, \tilde{u}'_X \in U_X, \tilde{o}_X \in O_{Op_X} \bullet stp_{Op_X}(u_X, i_X, \tilde{u}'_X, \tilde{o}_X) \wedge \\ &\qquad d^G_{X,Y}(\tilde{u}'_X, u'_Y) \leq \kappa\,(d^G_{X,Y}(u_X, u_Y) + d^{\ni}_{X,Y}(i_X, i_Y)) \wedge \\ &\qquad Out^{\Delta_O}_{Op_{X,Y}}(\tilde{o}_X, o_Y) \wedge EqEnbl_{X,Y}(\tilde{u}'_X, u'_Y)) \end{aligned} \tag{149}$$

In (149), $\kappa < (1 + \Delta_I/\Delta_G)^{-1}$. So our assertion that $\Delta_I = 0$, leads to $\kappa < 1$.

The initial value of $d^G_{X,Y}(u_X, u_Y)$ is 0 because both systems start at $(0, 0)$, so we conclude, by induction, that $d^G_{X,Y}(u_X, u_Y)$ must be 0 at each T_P interval boundary, since that is the only way to satisfy $d^G_{X,Y}(\tilde{u}'_X, u'_Y) \leq \kappa\,(d^G_{X,Y}(u_X, u_Y) + 0)$ for every interval. So this instance of Theorem 8.12 is applicable only to the execution where the state remains constantly at $(0, 0)$ without any non-zero $e?$ input.[26] We conclude that the no-origin instance of Theorem 8.12 is of limited interest for our case study.

If we now go to the with-origin case, the situation changes, since the revised definitions of distance functions permit them to focus on absolute magnitudes instead of differences between abstract and concrete values. If we take advantage of the flexibility afforded by the $(\eta, \alpha_X, \alpha_Y)$ parameters to redefine:

$$G^{\Delta_G}_{X,Y}(u_X, u_Y) \;\equiv\; \tfrac{1}{2}\,(\mu(u_X) + \mu(u_Y)) \leq \Delta_G = X_B \tag{150}$$

$$In^{\Delta_I}_{Op_{X,Y}}(i_X, i_Y) \;\equiv\; \tfrac{1}{2}\,(\mu(i_X) + \mu(i_Y)) \leq \Delta_I = E_B \tag{151}$$

we can deduce that Theorem 8.12 is applicable to our case if the dynamics of (129), (130), (131) satisfies the redefined constraints of the third line of (149). Working to leading order, expressed via

[25] If we were following the first version rather than the second, then the flexibility offered by the existential quantification in (148) would have been vital in allowing a choice of *ActContMch*_2.5 dynamics that exactly matched the *ActContMch*_1.01 dynamics.

[26] We ignore the bizzare possibility that the earthquake is precisely such that $d^G_{X,Y}(u_X, u_Y)$ is not identically zero, yet, returns to zero at each T_P interval boundary.

(131), and considering a full T_P interval, this is the case if:

$$\left(1 - \frac{\omega^2\,\tau^2}{2}\right) xx(\mathbb{t}_{\mathrm{L}}) + \left|\frac{a_X\,\tau^2}{2}\right| \leq \kappa\,(d^G_{X,Y}(u_X, u_Y) + d^{\ni}_{X,Y}(i_X, i_Y)) \tag{152}$$

where we have used the triangle inequality to ensure the inhomogeneous term of (131) contributes positively, and where $\tau = T_P$, $xx(\mathbb{t}_{\mathrm{L}}) = \Delta_G = X_B$, $\Delta_I = E_B$, $\kappa = (1 + \Delta_I/\Delta_G)^{-1}$, $d^G_{X,Y}(u_X, u_Y) = \Delta_G = X_B$, $d^{\ni}_{X,Y}(i_X, i_Y) = \Delta_I = E_B$, and where we have taken limiting values at all strict inequalities, justified by the desire to derive the extremal values permitted by (152). Evidently, the key factor in satisfying (152) is whether the additive inhomogeneous contribution on the left, can be dominated by the multiplicatively reduced inhomogeneous contribution on the right. Making the substitutions indicated leads to:

$$|\,a_X\,| \;\leq\; \omega^2\,X_B \qquad \text{or} \qquad \left|\frac{pp}{m} - e?\right| \;\leq\; \frac{X_B}{400\,T_P^2} \tag{153}$$

The large denominator in (153) places severe bounds on the magnitude of the earthquake that can be accommodated by the with-origins framework of Theorem 8.12 when applied to the *ActContMch_2.5* model in which there is no pp compensation in the interior of a T_P interval. We also observe that taking X_B as maximal before-value of $xx(\mathbb{t}_{\mathrm{L}})$ and respecting the constraint just derived results in a smaller before-value for $xx(\mathbb{t}_{\mathrm{L}})$ in the next interval, and so on. Starting with the known initial state of (0.0) would constrain the possibilities even further. In that sense, the with-origin version of Theorem 8.12 is not actually very useful, although it is more applicable than the no-origin version.

Asking that the invariants are maintained in the interior of a T_P interval, and not just at the interval boundaries, Policies 9.2 provide a way forward. We note that in the interior of a T_P interval, Policy 9.2.(1) will work, since the contracting nature of the dynamics implies that an already acceptable initial value of a bound will not be exceeded. Policy 9.2.(2) will not work, since the dynamics is concave rather than convex. Policy 9.2.(3) will work though, since the concave nature of the dynamics can be described by a monotonically decreasing quadratic function of time, as is clear from (152), generating a candidate for the required ϕ.

13.4.4 **Theorem 8.16.** When we progress to Theorem 8.16, the greater flexibility of the diverging/converging framework gives a little more scope for capturing the anticipated behaviour of the earthquake system in the real world. Definition 8.15 can be seen as combining a number of cases as in Theorem 8.12, but with different values of the κ parameter (called k now) in the diverging/converging criterion (67), together with suitable conditions to enable straightforward case analysis. Theorem 8.16 then instantiates this for two k values $0 < \kappa < 1 < K$, and presents two results, (a) and (b), each with a no-origin case and a with-origin case. The argument for the no-origin cases for both (a) and (b) is identical to that in Theorem 8.12, as the argument does not depend on the value of k.

For the with-origin cases, result (a) assumes each execution starts with $k = K$ steps. For us, this means the earthquake protection system starts exactly at the commencement of an earthquake, which is rather far fetched. So we will disregard result (a). (We note though that it speaks of an *Init*-constrained trace inclusion, to cope with the diverging $d^G_{X,Y}$ at the beginning of every considered execution.)

Result (b) assumes each execution starts with $k = \kappa$ steps, i.e. no ongoing earthquake at initialisation, which is much more plausible. If we choose (150) and (151) again for the with-origin case, we modify the relentless decrease in $d^G_{X,Y}$ of Theorem 8.12 by allowing for intermittent periods of increase via limited duration episodes of $k = K$ steps. However, since these would be assumed to

correspond to earthquakes, this version of events is also not entirely convincing from an application perspective.

We note that Theorem 8.16 gives us the freedom to separately choose input bounds for different k cases via the $d^{\ni}_{X,Y}(i_X, i_Y) \leq B_k$ clause of (67), so we do not have to commit to (151). This flexibility is reflected in the B_K and B_κ parameters of the trace inclusion data of result (b) of the theorem. Once this is understood, we can derive consequences analogous to (153) for this, slightly more complicated scenario.

13.4.5 **Theorem 8.20.** With Theorem 8.20, we have results that are more realistic with respect to the requirements of the earthquake protection application than earlier ones. The presence of a threshold value, below which the contracting nature of the protection system's dynamics is not insisted on, corresponds to the normal state of affairs in the long periods between earthquake episodes, when only minor environmental vibrations are detected.

Threshold effects aside, Theorem 8.20 follows the pattern of Theorem 8.16. There are, again, two results, (a) and (b), each with a no-origin case and a with-origin case. The no-origin cases are handled as before, as the arguments used are equally applicable.

For the with-origin cases, result (a) assumes each execution starts with $k = K$ steps, and yields an *Init*-constrained trace inclusion. Result (b) more realistically assumes a more peaceful start for each considered execution, and yields a trace inclusion parameterised by choices of input bounds B_K, B_κ, duration parameters M, n, threshold bounds B_T, D_T, and state bound Δ_G. As before, we can then derive consequences analogous to (153) for this, yet more complicated scenario, by combining the techniques that led to Theorem 8.20 and to (153).

13.4.6 **Theorem 7.2.** The discussion of last few sections, and specifically that in Section 13.4.5, allows us to recast our account of the relationship between *ActContMch*_1.01 and *ActContMch*_2.5 in the framework of Theorem 7.2. For the sets of executions considered in Theorems 8.16 and 8.20, we asserted that an excursion to larger amplitude x/xx values during an earthquake (diverging episode) was always followed by a restoration of smaller amplitude x/xx values (converging episode). Focusing on Theorem 8.20, we further envisaged long periods of below threshold values, before the next earthquake.

If we define a gluing relation between machines *ActContMch*_1.01 and *ActContMch*_2.5 by:

$$G_T((x, y), (xx, yy)) \;\equiv\; |\,x\,| + |\,xx\,| \leq D_T \tag{154}$$

where D_T is a (small) threshold value[27] adequate to accommodate the sum of the x/xx displacements during the quiescent periods between earthquakes, then with suitable W/D, the simulations captured by Theorem 8.20 can be described also as weak simulations according to Theorem 7.2. The earthquake episodes in such simulations are captured by bridging sequences (in the terminology of Section 6). We observe that the success of such a reinterpretation depends entirely on choosing different retrenchment data to describe the *ActContMch*_1.01 and *ActContMch*_2.5 relationship.

13.4.7 **Retrospective.** We close our account of the *ActContMch*_1.01 and *ActContMch*_2.5 relationship with a couple of final observations. Firstly, we did not try to cast the relationship in terms of *all* of the theorems in Section 8. Partly this was for economy, and partly it was because some of them, e.g. the ones dealing with unbounded behaviours, were less relevant to a system predicated on maintaining the $|x/xx| \leq X_B$ bound (and since all engineered systems have to assume limited ranges of parameters).

[27] The threshold D_T relevant to background noise, should not be confused with the threshold X_{th} in Fig. 7 relevant to judging whether or not to invoke active control, which is a much bigger value.

Secondly, we could avoid restricting to a set of executions **M** in those results that did so, by introducing suitable history variables into the models and by incorporating suitable conditions into the guards of events. However this is a rather artificial approach.

Thirdly, we deliberately avoided discussing one particular feature of the *ActContMch*_1.01 and *ActContMch*_2.5 relationship. This is that the extrapolation of the sampled values $x18$, $x19$, $e19$, to values at T_P in the *ActContMch*_2.5 model need not (indeed, generally will not) agree precisely with the exact values of x, y, e? values available in the *ActContMch*_1.01 model. This could cause a divergence of behaviours when x and xx are near X_{th} as one value could be below X_{th} and the other above. Our focus on G relationships based on absolute magnitudes allowed us to evade the issue. We postpone discussion of such effects to the next section.

13.5 *ActContMch*_2.5 and *ActContMch*_3

While the previous derivation step modelled discretisation in time, this section explores quantization of sensor and actuator values. This is introduced by the red $K_{\ldots}^{-1}\lfloor K_{\ldots}\ldots\rceil$ insertions in Fig. 7, which morph *ActContMch*_2.5 into *ActContMch*_3. Since both models use the same variable names, we revert to subscripts 2 and 3 to distinguish them.

One immediate consequence of quantization is that it becomes impossible to maintain exact equalities of state variables between the two models, e.g. $xx_2 = xx_3$, because a quantized answer to a calculation on quantized inputs will not be the same as the unquantized answer to the calculation on unquantized inputs, except by chance — we mentioned such effects already. However, they are easier to address in this section because the events of *ActContMch*_3 are just quantized versions of the events of *ActContMch*_2.5, i.e. the simulations needed consist of $(1,1)$ diagrams. Thus the analogue of the earlier (m,n) diagrams, e.g. (127), becomes:

$$\begin{array}{l} MONITOR_2 \fatsemi Sample\_18_2 \fatsemi MONITOR_2 \fatsemi Sample\_19_2 \fatsemi MONITOR_2 \fatsemi \mathsf{Last}_2 \\ \qquad\qquad\qquad\qquad \circledast \\ MONITOR_3 \fatsemi Sample\_18_3 \fatsemi MONITOR_3 \fatsemi Sample\_19_3 \fatsemi MONITOR_3 \fatsemi \mathsf{Last}_3 \end{array} \tag{155}$$

which can now be broken up into:

$$MONITOR_2 \circledast MONITOR_3 \tag{156}$$
$$Sample\_18_2 \circledast Sample\_18_3 \tag{157}$$
$$Sample\_19_2 \circledast Sample\_19_3 \tag{158}$$
$$\mathsf{Last}_2 \circledast \mathsf{Last}_3 \tag{159}$$

where (159) covers possibilities constructed from $\{PulseNo, PulseMaybe, PulseYesY, PulseYesE\}$.

With this insight we construct a retrenchment from *ActContMch*_2.5 to *ActContMch*_3, with data as follows:

- $\succcurlyeq \equiv \{\ MONITOR_2 \succcurlyeq MONITOR_3, Sample\_18_2 \succcurlyeq Sample\_18_3, Sample\_19_2 \succcurlyeq Sample\_19_3,$
  $PulseNo_2 \succcurlyeq PulseNo_3, PulseMaybe_2 \succcurlyeq PulseMaybe_3,$
  $PulseYesY_2 \succcurlyeq PulseYesY_3, PulseYesE_2 \succcurlyeq PulseYesE_3\ \} \cup$
  $\{\ PulseMaybe_2 \succcurlyeq PulseNo_3, PulseNo_2 \succcurlyeq PulseMaybe_3,$
  $PulseNo_2 \succcurlyeq PulseYesY_3, PulseYesY_2 \succcurlyeq PulseNo_3,$
  $PulseNo_2 \succcurlyeq PulseYesE_3, PulseYesE_2 \succcurlyeq PulseNo_3,$
  $PulseMaybe_2 \succcurlyeq PulseYesY_3, PulseYesY_2 \succcurlyeq PulseMaybe_3,$
  $PulseMaybe_2 \succcurlyeq PulseYesE_3, PulseYesE_2 \succcurlyeq PulseMaybe_3,$
  $PulseYesY_2 \succcurlyeq PulseYesE_3, PulseYesE_2 \succcurlyeq PulseYesY_3\ \}$ (160)

The second term in (160) encompasses two sorts of case. The first sort is generated by all the cases where a case distinction in the two models is decided on the value of an inequality, which could, because of a small quantisation discrepancy between the models, be flipped into different actions. The second sort is when a historic occurrence of the former sort has led to a divergence between the models. In such cases no combination of simultaneous occurrences of different Last actions in the two models can be excluded.

For the gluing relation, we have the obvious identity:

- $G_{2,3}(xx_2, yy_2, x18_2, x19_2, y19_2, e19_2, xx_3, yy_3, x18_3, x19_3, y19_3, e19_3) \;\equiv$
$$xx_2 = xx_3 \wedge yy_2 = yy_3 \wedge x18_2 = x18_3 \wedge x19_2 = x19_3 \wedge y19_2 = y19_3 \wedge e19_2 = e19_3 \qquad (161)$$

Before discussing the events, we make the following observations.

- For simplicity, we assume that $K_{xs} = K_{es}$, and that the maximum rounding error caused by any $K_{\ldots}^{-1} \lfloor K_{\ldots} \ldots \rceil$ insertion is $\delta \in \mathbb{R}$.
- **"***e*/*e*?-Eq**"** is as in (118), with adjusted subscripts.
- **"**P-Dsc-2**"** is as **"**P-Dsc'-2**"** in (120), but without the heavy apostrophe decorating the $x19$, $y19$, $e19$ variables. These indicated the after-values in the interior of the (m, n) diagrams of Sections 13.2 and 13.3. But those values are the before-values of the various Last events needed here. **"**P-Dsc-3**"** denotes the same thing but where the model subscript is 3 rather than 2, referring to *ActContMch*_3.
- In formulating the data for the event retrenchments below, we focus on some key facts, and in the comments, we hint at the main idea in the proof where needed. Evidently additional facts could easily be included in the retrenchment data, if desired. This particularly applies to the more detailed criteria needed to separate the *YesY* cases from the *YesE* cases (indicated by **"***Y*.gtr.*E***"** and **"***E*gtr.*Y***"** in Sections 13.2 Section 13.3), which we have omitted, leading to some duplication among the retrenchment data. The same could be said regarding finer grained distinctions arising from the different possibilities mentioned immediately after (160), which would, however, require the subdivision of the events of the two machines, leading to additional syntactic complication.

The event retrenchment data are now:

MONITOR$_2 \succcurlyeq$ *MONITOR*$_3$

- $W_{MONITOR_2 \succcurlyeq MONITOR_3}(\ldots) \;\equiv\; |xx_2(\mathbb{t}_\mathrm{L})| \le X_B \wedge |xx_3(\mathbb{t}_\mathrm{L})| \le X_B \wedge$ **"***e*/*e*?-Eq**"** $\wedge$
$$\mathbb{t}_\mathrm{R} - \mathbb{t}_\mathrm{L} \le 18T_P/20 \wedge$$
$$\left| x19_2(\mathbb{t}_\mathrm{L}) \left(1 - \omega^2 T_P^2/2\right) + y19_2(\mathbb{t}_\mathrm{L})\, T_P\left(1 - \zeta\, \omega\, T_P\right) - e19_2(\mathbb{t}_\mathrm{L})\, T_P^2/2 \right| \le X_B \wedge$$
$$\left| x19_3(\mathbb{t}_\mathrm{L}) \left(1 - \omega^2 T_P^2/2\right) + y19_3(\mathbb{t}_\mathrm{L})\, T_P\left(1 - \zeta\, \omega\, T_P\right) - e19_3(\mathbb{t}_\mathrm{L})\, T_P^2/2 \right| \le X_B \qquad (162)$$
- $D_{MONITOR_2 \succcurlyeq MONITOR_3}(\ldots) \;\equiv\; \big(\, \forall \tau \in (\mathbb{t}_\mathrm{L} \ldots \mathbb{t}_\mathrm{R}) \bullet$
$$\left| x19_2(\tau) \left(1 - \omega^2 T_P^2/2\right) + y19_2(\tau)\, T_P\left(1 - \zeta\, \omega\, T_P\right) - e19_2(\tau)\, T_P^2/2 \right| \le X_B \wedge$$
$$\left| x19_3(\tau) \left(1 - \omega^2 T_P^2/2\right) + y19_3(\tau)\, T_P\left(1 - \zeta\, \omega\, T_P\right) - e19_3(\tau)\, T_P^2/2 \right| \le X_B \,\big) \wedge$$
$$\big(\, \forall \tau \in (\mathbb{t}_\mathrm{L} \ldots \mathbb{t}_\mathrm{R}) \bullet |e19(\tau)| \le E_{th} \Rightarrow |xx_2(\tau) - xx_3(\tau)| < |xx_2(\mathbb{t}_\mathrm{L}) - xx_3(\mathbb{t}_\mathrm{L})| \,\big) \qquad (163)$$

In a 1-1 retrenchment context, the longest continuous event duration is from the beginning of a T_P interval up to the *Sample_18* events. The last two assertions in (162) hold because it is the responsibility of the preceding T_P interval to apply a jolt to the $yy_{2/3}$ variables (if necessary) in order not only that those assertions do hold at $\mathbb{t}_\mathrm{L}$, but that the corresponding inequalities hold throughout the current T_P interval

too, as asserted in (163). The last assertion in (163) states that provided the external earthquake disturbance is smaller than a small threshold E_{th}, then the dynamics will be contracting, as discussed earlier.

Sample_18$_2 \succcurlyeq$ *Sample_18*$_3$

- $W_{Sample\_18_2 \succcurlyeq Sample\_18_3}(\ldots) \equiv |xx_2| \leq X_B \wedge |xx_3| \leq X_B \wedge$ **"**$e/e?$-Eq**"** (164)
- $D_{Sample\_18_2 \succcurlyeq Sample\_18_3}(\ldots) \equiv$ **"**$xx_{2/3}$-Eq**"** $\wedge$ **"**$yy_{2/3}$-Eq**"** $\wedge$
  $\big|\,|x18'_2 - x18'_3| - |x18_2 - x18_3|\,\big| \leq \delta$ (165)

"$xx_{2/3}$-Eq**"** denotes $xx'_2 = xx_2 \wedge xx'_3 = xx_3$; **"**$yy_{2/3}$-Eq**"** denotes $yy'_2 = y_2 \wedge yy'_3 = yy_3$.

Sample_19$_2 \succcurlyeq$ *Sample_19*$_3$

- $W_{Sample\_19_2 \succcurlyeq Sample\_19_3}(\ldots) \equiv |xx_2| \leq X_B \wedge |xx_3| \leq X_B \wedge$ **"**$e/e?$-Eq**"** (166)
- $D_{Sample\_19_2 \succcurlyeq Sample\_19_3}(\ldots) \equiv$ **"**$xx_{2/3}$-Eq**"** $\wedge$ **"**$yy_{2/3}$-Eq**"** $\wedge$
  $\big|\,|x19'_2 - x19'_3| - |x19_2 - x19_3|\,\big| \leq \delta \wedge$
  $\big|\,|y19'_2 - y19'_3| - |y19_2 - y19_3|\,\big| \leq 40\,\delta/T_P \wedge$
  $\big|\,|e19'_2 - e19'_3| - |e19_2 - e19_3|\,\big| \leq \delta$ (167)

  Variable $y19'_3$ could be vulnerable to two rounding errors, rescaled by $20/T_P$.

PulseNo$_2 \succcurlyeq$ *PulseNo*$_3$

- $W_{PulseNo_2 \succcurlyeq PulseNo_3}(\ldots) \equiv |xx_2| \leq X_{th} < X_B \wedge |xx_3| \leq X_{th} < X_B$ (168)
- $D_{PulseNo_2 \succcurlyeq PulseNo_3}(\ldots) \equiv$ **"**$xx_{2/3}$-Eq**"** $\wedge$ **"**$yy_{2/3}$-Eq**"** (169)

PulseMaybe$_2 \succcurlyeq$ *PulseMaybe*$_3$

- $W_{PulseMaybe_2 \succcurlyeq PulseMaybe_3}(\ldots) \equiv X_{th} \leq |xx_2| \leq X_B \wedge X_{th} \leq |xx_3| \leq X_B \wedge$
  "P-Dsc-2**"** $\leq X_B \wedge$ **"**P-Dsc-3**"** $\leq X_B$ (170)
- $D_{PulseMaybe_2 \succcurlyeq PulseMaybe_3}(\ldots) \equiv$ **"**$xx_{2/3}$-Eq**"** $\wedge$ **"**$yy_{2/3}$-Eq**"** (171)

PulseYesY$_2 \succcurlyeq$ *PulseYesY*$_3$ and *PulseYesE*$_2 \succcurlyeq$ *PulseYesE*$_3$

- $W_{PulseYesY/E_2 \succcurlyeq PulseYesY/E_3}(\ldots) \equiv X_{th} \leq |xx_2| \leq X_B \wedge X_{th} \leq |xx_3| \leq X_B \wedge$
  "P-Dsc-2**"** $\geq X_B \wedge$ **"**P-Dsc-3**"** $\geq X_B$ (172)
- $D_{PulseYesY/E_2 \succcurlyeq PulseYesY/E_3}(\ldots) \equiv$ **"**$xx_{2/3}$-Eq**"** $\wedge\, |yy'_2 - yy'_3| \leq \frac{1}{2}|yy_2 - yy_3| + \delta$ (173)

  Variables yy_2 and yy_3 are both reduced in magnitude by at least half. The W and D relations for the *YesY* and *YesE* cases are identical.

PulseNo$_2 \succcurlyeq$ *PulseMaybe*$_3$

- $W_{PulseNo_2 \succcurlyeq PulseMaybe_3}(\ldots) \equiv |xx_2| \leq X_{th} < X_B \wedge X_{th} \leq |xx_3| \leq X_B \wedge$
  "P-Dsc-3**"** $\leq X_B$ (174)
- $D_{PulseNo_2 \succcurlyeq PulseMaybe_3}(\ldots) \equiv$ **"**$xx_{2/3}$-Eq**"** $\wedge$ **"**$yy_{2/3}$-Eq**"** (175)

PulseMaybe$_2 \succcurlyeq$ *PulseNo*$_3$

- $W_{PulseMaybe_2 \succcurlyeq PulseNo_3}(\ldots) \equiv X_{th} \leq |xx_2| \leq X_B \wedge |xx_3| \leq X_{th} < X_B \wedge$
  "P-Dsc-2**"** $\leq X_B$ (176)
- $D_{PulseMaybe_2 \succcurlyeq PulseNo_3}(\ldots) \equiv$ **"**$xx_{2/3}$-Eq**"** $\wedge$ **"**$yy_{2/3}$-Eq**"** (177)

$\underline{PulseNo_2 \succcurlyeq PulseYesY_3}$ and $\underline{PulseNo_2 \succcurlyeq PulseYesE_3}$

- $W_{PulseNo_2 \succcurlyeq PulseYesY/E_3}(\ldots) \equiv |\,xx_2\,| \leq X_{th} < X_B \wedge X_{th} \leq |\,xx_3\,| \leq X_B \wedge$ “P-Dsc-3” $\geq X_B$ (178)
- $D_{PulseNo_2 \succcurlyeq PulseYesY/E_3}(\ldots) \equiv$ “$xx_{2/3}$-Eq” $\wedge\ yy_2' = yy_2 \wedge |\,yy_3'\,| \leq \frac{1}{2}|\,yy_3\,| + \delta$ (179)

$\underline{PulseYesY_2 \succcurlyeq PulseNo_3}$ and $\underline{PulseYesE_2 \succcurlyeq PulseNo_3}$

- $W_{PulseYesY/E_2 \succcurlyeq PulseNo_3}(\ldots) \equiv X_{th} \leq |\,xx_2\,| \leq X_B \wedge |\,xx_3\,| \leq X_{th} < X_B \wedge$ “P-Dsc-2” $\geq X_B$ (180)
- $D_{PulseYesY/E_2 \succcurlyeq PulseNo_3}(\ldots) \equiv$ “$xx_{2/3}$-Eq” $\wedge\ |\,yy_2'\,| \leq \frac{1}{2}|\,yy_2\,| \wedge yy_3' = yy_3$ (181)

$\underline{PulseMaybe_2 \succcurlyeq PulseYesY_3}$ and $\underline{PulseMaybe_2 \succcurlyeq PulseYesE_3}$

- $W_{PulseMaybe_2 \succcurlyeq PulseYesY/E_3}(\ldots) \equiv X_{th} \leq |\,xx_2\,| \leq X_B \wedge X_{th} \leq |\,xx_3\,| \leq X_B \wedge$ “P-Dsc-2” $\leq X_B \wedge$ “P-Dsc-3” $\geq X_B$ (182)
- $D_{PulseMaybe_2 \succcurlyeq PulseYesY/E_3}(\ldots) \equiv$ “$xx_{2/3}$-Eq” $\wedge\ yy_2' = yy_2 \wedge |\,yy_3'\,| \leq \frac{1}{2}|\,yy_3\,| + \delta$ (183)

$\underline{PulseYesY_2 \succcurlyeq PulseMaybe_3}$ and $\underline{PulseYesE_2 \succcurlyeq PulseMaybe_3}$

- $W_{PulseYesY/E_2 \succcurlyeq PulseMaybe_3}(\ldots) \equiv X_{th} \leq |\,xx_2\,| \leq X_B \wedge X_{th} \leq |\,xx_3\,| \leq X_B \wedge$ “P-Dsc-2” $\geq X_B \wedge$ “P-Dsc-3” $\leq X_B$ (184)
- $D_{PulseYesY/E_2 \succcurlyeq PulseMaybe_3}(\ldots) \equiv$ “$xx_{2/3}$-Eq” $\wedge\ |\,yy_2'\,| \leq \frac{1}{2}|\,yy_2\,| \wedge yy_3' = yy_3$ (185)

$\underline{PulseYesY_2 \succcurlyeq PulseYesE_3}$ and $\underline{PulseYesE_2 \succcurlyeq PulseYesY_3}$

- $W_{PulseYesY/E_2 \succcurlyeq PulseYesE/Y_3}(\ldots) \equiv X_{th} \leq |\,xx_2\,| \leq X_B \wedge X_{th} \leq |\,xx_3\,| \leq X_B \wedge$ “P-Dsc-2” $\geq X_B \wedge$ “P-Dsc-3” $\geq X_B$ (186)
- $D_{PulseYesY/E_2 \succcurlyeq PulseYesE/Y_3}(\ldots) \equiv$ “$xx_{2/3}$-Eq” $\wedge\ |\,yy_2' + yy_3'\,| \leq \frac{1}{2}|\,yy_2 + yy_3\,| + \delta$ (187)

The above technical details have been kept reasonably simple for the sake of brevity. However, the remark concerning the last line of (163) indicates that a finer case analysis, with more detailed conclusions in the D relations would not be impossible. We refrain from exploring such elaborations.

13.6 Coda

The preceding sections covered a rich portfolio of relationships between the various models we introduced in Section 12. Fig. 9 summarises the situation. This variety vividly illustrates that the individual steps in a formal development *must be chosen by humans*. This contrasts with the picture often seen in more conventional refinement developments, which are often presented as if there were little room for credible alternative approaches, a picture reinforced when the application is relatively simple.

The variety spoken of is the more keenly felt in the hybrid world, where so many techniques, particularly ones implementing approximations of various kinds, can be pursued in various combinations, and to as high an order as one chooses, forcing one to be selective. The hybrid case also underlines the fact that (even in conventional, discrete cases) *human beings* have to decide:

- *what* goes into the models of a development,
- *which* relationships between the models are formalised,
- *to what extent* those formal relationships are mechanised,
- *how much* reliance is placed on the outcome of such efforts,
- *how much* residual risk resides in matters *not* explored by these means.

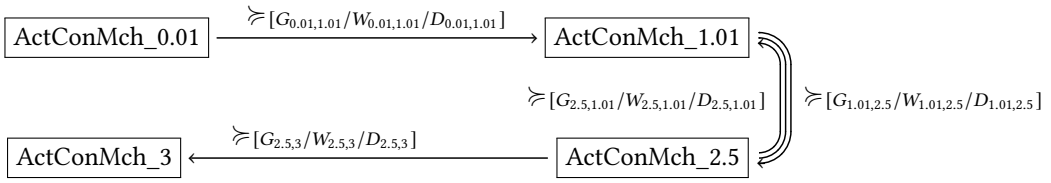

Fig. 9. The earthquake damage prevention active control system development hierarchy discussed in Section 13, including the relevant formal development steps. Between *ActConMch*_1 and *ActConMch*_2.5, the downward arrow depicts the $[G_{1,2.5}/W_{1,2.5}/D_{1,2.5}]$ retrenchment (first version), the upward arrow depicts the $[G_{2.5,1}/W_{2.5,1}/D_{2.5,1}]$, retrenchment (second version), and the middle line refers to the simulations of the third version.

14 EARTHQUAKE PROTECTION AS GRADED DEVELOPMENT SYSTEM

Having explored a number of options in the previous section, we can now put them together to assemble, in a number of different ways, a GDS as discussed in Section 10. While Fig. 9 summarised the relationships we explored in Section 13, Fig. 10 shows the partial orders that can be constructed from these, using the simplified labels. Thus, in Fig. 10.(a) we see the straightforward development consisting of successive retrenchments from *ActContMch*_0.01 to *ActContMch*_1.01, then to *ActContMch*_2.5, and then to *ActContMch*_3, making use of the first version of the *ActContMch*_2.5 to *ActContMch*_3 relationship. The formal relationship between *ActContMch*_0.01 and *ActContMch*_3 that this leads to can be described by several appropriate (m, n) diagrams, covering the range of options indicated by Last. The shapes would all be the same simple adaptation of (112), namely:

$$\begin{array}{c} MONITOR_0 \\ \circledast \\ MONITOR_3 \fatsemi Sample\_18_3 \fatsemi MONITOR_3 \fatsemi Sample\_19_3 \fatsemi MONITOR_3 \fatsemi \mathsf{Last}_3 \end{array} \tag{188}$$

In each case, this arises from (112), firstly: by extending above, by composing with a $(1, 2)$ diagram that deals with the $MONITOR_0 \circledast MONITOR_1 \fatsemi MoSkip_1$ development step; and secondly: by extending below with a $(5, 5)$ diagram built by abutting the five $(1, 1)$ diagrams that bridge between the *ActContMch*_2.5 and *ActContMch*_3 versions of the events in the bottom line of (188).

Regarding $G_{0,3}$ and the various $W_{0,3}$ and $D_{0,3}$ relations that arise, in each case they are created by routine relational composition: we write down the conjunction of the relevant relations from the component retrenchments elaborated earlier, taking care that the variable names are consistent across all needed component relations, and we then existentially quantify all the internal variables. We omit the details.

Fig. 10.(b) shows what happens when we consider the second and third versions of the relationship between *ActContMch*_2.5 and *ActContMch*_3 — the orientation of the middle link is switched, and the relations belonging to it are composed in transposed manner (with the relations belonging to links above and below). Since, for those relations, the main difference is between $G_{1,2}$ (an inverse projection), and $G_{2,1}$ (a universal relation on $|x/xx| \leq X_B$ and inverse projection on the rest), and since aside from the caveats discussed in Section 13.3 just before (13.3), the other relations are the same as those used for Fig. 10.(a), it follows that the (m, n) diagrams that are derived are also very similar to those for Fig. 10.(a).

Fig. 10.(a)-(b) reflect what is derived in Section 10 and shown in Fig. 9. Fig. 10.(c) shows what would happen if we pursued the possibility of developing the approximate models to higher order,

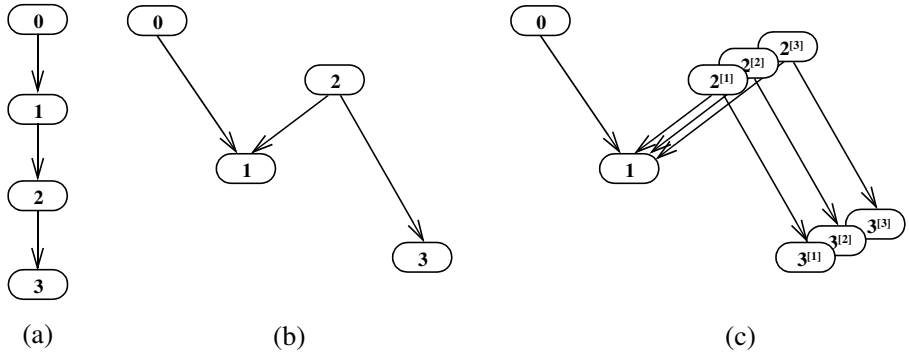

Fig. 10. Diagrammatic of development strategies for the earthquake protection system. (a) A development using the first version of the middle step. (b) A development using the second version of the middle step (also appropriate for the third version). (c) A hypothetical development icorporating higher order versions of *ActContMch*_2.5 and *ActContMch*_3.

indicated by the [1], [2], [3] superscripts. There would be arrows $2^{[1]} \longrightarrow 2^{[2]}$ and $2^{[2]} \longrightarrow 2^{[3]}$, which we have not shown. Likewise arrows $3^{[1]} \longrightarrow 3^{[2]}$ and $3^{[2]} \longrightarrow 3^{[3]}$. If elaborated in full detail, the data for such arrows would typically feature approximate relationships, which we have indicated earlier using notations like $*^{\mathrm{o}((\omega\tau)^k)}$.

Regarding the various V sets discussed in Section 10.1, we can comment briefly on V and on $VInv$. Let $\gamma_{[0,1,2,3]}$ denote the path in Fig. 10.(a), and let $\gamma_{[0,\overline{1,2},3]}$ denote the path in Fig. 10.(b). In fact, for $\gamma \in \{\gamma_{[0,1,2,3]}, \gamma_{[0,\overline{1,2},3]}\}$, $V_\gamma = VInv_\gamma$. This is because the $|xx| \leq X_B$ invariant has been built into $G_{1,2}$ and $G_{2,1}$, and the identities and projections of the other basic links propagate this to the 0 and 3 ends of $\gamma_{[0,1,2,3]}$ and $\gamma_{[0,\overline{1,2},3]}$, regardless of whether the invariants are checked at the ends of the paths.

One possibility we have not followed above, but which is suggested by the later results we explored, particularly the ones which singled out the benign no-current-earthquake behaviours, is to craft the development starting with purely benign behaviour, and to then introduce suitable retrenchments to incorporate the effects of the large displacements caused by earthquakes (and the system's consequent countermoves). Although we do not explore this possibility in technical detail, Fig. 11 illustrates how such a development path might go. In Fig. 11.(a) we see a depiction of an initial development, without earthquakes, patterned after Fig. 10.(a). Then in Fig. 11.(b) we see a series of

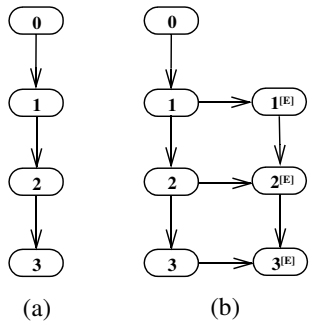

Fig. 11. A hypothetical development of the earthquake protection system based on introducing earthquakes late. (a) Initial, earthquake-free development. (b) The development following the addition of retrenchments to cater for earthquakes in *ActContMch*_1.01, *ActContMch*_2.5 and *ActContMch*_3.

retrenchments that introduce earthquake handling into *ActContMch*_1.01, *ActContMch*_2.5 and *ActContMch*_3, shown by the [E] superscript.

15 DISCUSSION

In this section we broaden the context of the discussion beyond the technical details of the previous sections, and we also examine related work.

15.1 Wider Contextual Considerations

Our focus in the previous sections was on bridging the impasse between the world in which desired system requirements were most eloquently articulated, and the world in which they were implemented, using techniques derived from model based formal refinement, and concentrated on the case of hybrid and cyber-physical systems, in which these issues are particularly vexing.

One way of tackling such vexing issues is to sidestep them, by drawing a sharp boundary between those aspects conveniently dealt with using model based techniques, and the rest. Thus, in the case of hybrid and cyber-physical systems, the structural concerns in such systems are typically clean and discrete, and lend themselves well to logically based formal refinement approaches, while the continuous elements are relegated to one or more parts of the system into which formal techniques do not intrude. Running counter to this is the argument that, in fact, the continuous elements are normally given via ODE systems, and an ODE is a specification mechanism *par excellence*, in that it is a precise but implicit definition of expected behaviour, the explicit consequences of which are delegated to an ODE solution mechanism (assuming one is available). Viewed thus, when formal techniques wash their hands of the continuous elements, it is a kind of abrogation of responsibility.

Of course the nub of the problem is that (in the overwhelming proportion of cases) ODE systems are implementable only approximately. If we stick to the ideal world of mathematics, it is easy to argue that a solution to an ODE (which definitively exists under suitable assumptions, albeit non-executably) is a refinement of the behaviour specified by the ODE. As soon as we insist on executability though, the approximations involved in numerical algorithms break the clean criteria of typical refinement theories. However, rather than abandoning the challenge, we ought to take this as a spur to developing theories more applicable to the situation in question. It is our contention that the framework developed in this paper is a step in this direction, and some specific points relevant to our particular approach are worth elaborating.

A first point is that although the idea of exerting control via a series of 'delta functions', acting impulsively, arises naturally enough in engineering contexts, the counterpart within model based formal refinement is rather unnatural, and has to be handled with care. Delta functions can be integrated to yield step functions. The discontinuities that these exhibit can be handled well enough via instantaneous state update, but care has to be exercised in keeping the dimensions of different state variables consistent with physical theory.

A second point concerns the fact that in conventional engineering, approximate calculations that neglect higher order phenomena are often used for obtaining acceptable results quickly and efficiently, putting aside the fact that approximations are frequently the only route to any answer at all, given the limited reach of analytical techniques. Reconciling this fact with the precision that is characteristic of refinement based approaches requires ingenuity, and is often supported by using additional properties such as contraction and other convergence phenomena.

A third point extends the previous one, namely, that even in situations where analytical techniques can deliver precise answers in principle, implementation using conventional devices will invariably rely on discretisations and quantisations of various kinds, each of which brings with it a degree of imprecision and the kind of threshold crossing discrepancies we dealt with above. And whereas the previous issue can be tackled by formulating more ingenious notions of correspondence between

system models, addressing the latter requires a judicious balance between delving into detail and exploiting more global properties that can cover all the sources of low level imprecision that may arise.

15.2 Related Work

The hybrid and cyber-physical systems field has been under investigation for several decades by now, early works being e.g. [6, 46, 47]. This is also witnessed by the longevity of the *Hybrid Systems: Computation and Control* series of international conferences [50], these days absorbed into [36]. Many of the earlier approaches, and especially the tools that support the relevant methodologies are surveyed in [28]. The wide applicability of the techniques explored can be seen in [38]. By now, a number of texts have appeared, with [5, 56] being quite heavily biased towards discrete systems techniques. A theoretically based overview of earlier work is to be found in [75]. More recent work is surveyed in [37, 69].

In [63] we can find an extensive review of exact solutions for ODEs, while [44, 45] explore what is needed when no exact solutions are possible, this being the majority of the time.

The fact that it is often not possible to solve a hybrid/cyber-physical system exactly is not the insuperable obstacle it might seem to be. Often it is sufficient to know that a system will stay in a safe region of the state space indefinitely, as we often did above, without knowing the precise dynamics. When it is sufficient for the system to stay in a target region of the state space, various kinds of 'helper functions' may be employed to gain assurance, and these can be related to our use of metric and contractive concepts.

Variant functions are familiar from the classical discrete programming world [7, 35, 49]. To help control the behaviour of recursions and unbounded iterations, a variant function (of the state) is required to be decreased by each iteration's state change. When the variant function takes values in a well founded set, this gives a guarantee of termination.

Lyapunov functions are well known from continuous control theory [43, 48, 72]. To help establish stability, the flow defined by the dynamics is required to decrease the Lyapunov function (of the state), this being easier to ascertain than to argue about the flow itself. The Lyapunov function has an easily identified minimum, which coincides with a stable fixed point of the dynamics.

Barrier functions have become a familiar technique for establishing safety in the hybrid systems world [32, 55, 64]. They are required to have one sign (positive say) in the unsafe region, and to have the other sign (negative) in the set of initial states. Provided the barrier function is decreased by the flow defined by the continuous dynamics and is also decreased by each discrete state change, the unsafe region can never be reached.

The common feature of all of these techniques, and of our use of metric ideas, is the demand that the relevant quantity is decreased at each step of the dynamics to guarantee the desired property of interest. This idea, and the fact that it can handle situations in which exact solutions are not available is prominent in [75], as well as in [40–42]. This aspect will not vanish from reasoning about complex systems in the forseeable future.

Regarding the proof based approach of this paper, we have already discussed Hybrid Event-B [12, 13], of which the formalism of this paper is a variation. Other approaches targeted at proof include Hybrid CSP together with the tools that support it [46, 58, 82]. Another is the dynamic logic approach of Platzer along with the KeYmaera tool [61, 62, 74].

16 CONCLUSIONS

In this rather substantial paper, we have taken on the challenge of reappraising the original retrenchment notion in the light of the experience gained since its introduction, in order to improve its ability to confront difficult requirements scenarios within a formal framework compatible with

refinement. We saw that the looser connection between the data of a retrenchment and the trace inclusion property, gave rise to a wealth of possibilities for describing the relationship between models in a formal development pathway, especially regarding simulation. We saw that regarding recovery to a notion of simulability, metric ideas (allied in the previous section to variants and Lyapunov functions, etc.) were indispensable.

With this ground prepared, we embarked on a lengthy examination of the earthquake protection system development of [11]. This case study contains many requirements issues that are, *prima facie*, awkward to deal with within conventional refinement frameworks, as pointed out already in [11]. The original development in [11] was subjected to scrutiny from the many different vantage points developed here. Thus, the effort expended on the case study in this paper, occupying almost half of it, is well merited in the author's opinion. Generally, it was found that the theoretical notions in the earlier part of the paper were well able to handle what was needed for the case study.

The wider aim of the detailed scrutiny of the difficult requirements issues we encountered above is to make available to the wider system and software engineering an approach to handling such issues that offers some particular desirable characteristics. Firstly, it is intended to be more widely applicable than the well established refinement approach (since refinement, by itself, often cannot handle these issues 'as they come'). Secondly, it is intended to offer more flexibility, while retaining a useful degree of rigour, than refinement might (since refinement, even when applicable, can demand a rigidity of approach that may not be possible to accommodate in a realistic engineering environment). Thirdly, it is intended to be compatible with refinement, as was illustrated well in the case study, so that it is not necessary to deprive oneself of the benefits of refinement and of the stronger guarantees that it offers when that can be achieved. Fourthly, a major aim in formulating retrenchment was to retain the 'challenge/response' nature of the typical refinement approach. The approach of positing system properties (in the invariants) as a challenge, to be followed by the response of discharging system generated POs that guaranteed them, has been a major benefit of refinement approaches. So it was a deliberate design decision to craft the retrenchment notion to share similar characteristics, albeit that the properties in question were generated by the user to a greater extent than was the case with refinement.

A final, important, consequence of all this is that the deliberate similarity in approach to what has gone before with refinement, is intended to make tool support for retrenchment an easy adaptation and extension of tools for refinement, the easier to get the benefits of both, especially when departures from refinement are relatively limited in the development of a given application. It is to be hoped that the results of this paper will spur the wider adoption of the approach and the creation of tools along the lines suggested.

REFERENCES

[1] M. Abadi and L. Lamport. 1991. The Existence of Refinement Mappings. *Theor. Comp. Sci.* 82 (1991), 253–284.
[2] J.-R. Abrial. 1996. *The B-Book: Assigning Programs to Meanings.* Cambridge University Press.
[3] J.-R. Abrial. 2010. *Modeling in Event-B: System and Software Engineering.* Cambridge University Press.
[4] N. Ahmed. 2006. *Dynamic Systems and Control With Applications.* World Scientific.
[5] R. Alur. 2015. *Principles of Cyberphysical Systems.* MIT Press.
[6] R. Alur, C. Courcoubetis, T. Henzinger, and P-H. Ho. 1993. Hybrid Automata: An Algorithmic Approach to the Specification and Verification of Hybrid Systems. In *Proc. Workshop on Theory of Hybrid Systems (LNCS)*, Vol. 736. Springer, 209–229.
[7] K. Apt. 1981. Ten Years of Hoare's Logic: A Survey Part I. *A.C.M. Trans. Prog. Lang. Sys.* 3 (1981), 431–483.
[8] R. Banach. 1994. Regular Relations and Bicartesian Squares. *Theor. Comp. Sci.* 129 (1994), 187–192.
[9] R. Banach. 1995. On Regularity in Software Design. *Sci. Comp. Prog.* 24 (1995), 221–248.
[10] R. Banach. 2015. Model Based Refinement and the Design of Retrenchments. *J. Soft Comp. Soft. Eng.* 5 (2015), 31–54.
[11] R. Banach and J. Baugh. 2018. A Simple Hybrid Event-B Model of an Active Control System for Earthquake Protection. In *Proc. Susan Stepney Festschrift (Emergence, Complexity, Computation)*, Vol. 35. Springer, 157–194.

[12] R. Banach, M. Butler, S. Qin, N. Verma, and H. Zhu. 2015. Core Hybrid Event-B I: Single Hybrid Event-B Machines. *Sci. Comp. Prog.* 105 (2015), 92–123.

[13] R. Banach, M. Butler, S. Qin, and H. Zhu. 2017. Core Hybrid Event-B II: Multiple Cooperating Hybrid Event-B Machines. *Sci. Comp. Prog.* 139 (2017), 1–35.

[14] R. Banach and C. Jeske. 2015. Retrenchment and Refinement Interworking: the Tower Theorems. *Math. Struc. Comp. Sci.* 25 (2015), 135–202.

[15] R. Banach, C. Jeske, and M. Poppleton. 2008. Composition Mechanisms for Retrenchment. *J. Log. Alg. Prog.* 75 (2008), 209–229.

[16] R. Banach, C. Jeske, M. Poppleton, and S. Stepney. 2006. Retrenching the Purse: Finite Exception Logs, and Validating the Small. In *Proc. NASA/IEEE SEW-30*. IEEE, 234–245.

[17] R. Banach, C. Jeske, M. Poppleton, and S. Stepney. 2006. Retrenching the Purse: Hashing Injective CLEAR Codes, and Security Properties. In *Proc. ISOLA-06*. IEEE, 82–90.

[18] R. Banach, C. Jeske, M. Poppleton, and S. Stepney. 2007. Retrenching the Purse: The Balance Enquiry Quandary, and Generalised and (1,1) Forward Refinements. *Fund. Inf.* 77 (2007), 29–69.

[19] R. Banach and M. Poppleton. 1998. Retrenchment: An Engineering Variation on Refinement. In *Proc. B-98*, Vol. 1393. Springer, LNCS, 129–147.

[20] R. Banach, M. Poppleton, C. Jeske, and S. Stepney. 2005. Retrenching the Purse: Finite Sequence Numbers and the Tower Pattern. In *Proc. FM 2005*, Vol. 3582. Springer, LNCS, 382–398.

[21] R. Banach, M. Poppleton, C. Jeske, and S. Stepney. 2007. Engineering and Theoretical Underpinnings of Retrenchment. *Sci. Comp. Prog.* 67 (2007), 301–329.

[22] M. Bardi and I. Capuzzo-Dolcetta. 2008. *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations.* Birkhauser.

[23] M. Barr and C. Wells. 1990. *Category Theory for Computing Science.* Prentice-Hall.

[24] E. Boiten and J. Derrick. 2005. Formal Program Development with Approximations. In *Proc. ZB-05*, Vol. 3455. Springer, LNCS, 374–392.

[25] F. Borceux. 1994. *Handbook of Categorical Algebra, Vols I-III.* Cambridge University Press.

[26] E. Börger. 2003. The ASM Refinement Method. *Form. Asp. Comp.* 15 (2003), 237–257.

[27] E. Börger and R.F. Stärk. 2003. *Abstract State Machines. A Method for High Level System Design and Analysis.* Springer.

[28] L. Carloni, R. Passerone, A. Pinto, and A. Sangiovanni-Vincentelli. 2006. Languages and Tools for Hybrid Systems Design. *Foundations and Trends in Electronic Design Automation* 1 (2006), 1–193.

[29] C. Chicone. 2006. *Ordinary Differential Equations with Applications* (2nd ed.). Springer.

[30] A. Chopra. 2015. *Dynamics of Structures: Theory and Applications to Earthquake Engineering* (4th. ed.). Pearson.

[31] F. Clarke. 2013. *Functional Analysis, Calculus of Variations and Optimal Control.* Springer.

[32] L. Dai, T. Gan, B. Xia, and N. Zhan. 2017. Barrier Certificates Revisited. *J. Symb. Comp.* 80 (2017), 62–86.

[33] W.-P. de Roever and K. Engelhardt. 1998. *Data Refinement: Model-Oriented Proof Methods and their Comparison.* Cambridge University Press.

[34] J Derrick and E Boiten. 2001. *Refinement in Z and Object-Z: Foundations and Advanced Applications.* Springer-Verlag UK.

[35] E. Dijkstra. 1976. *A Discipline of Programming.* Prentice-Hall.

[36] ESW. Embedded Systems Week Conferences.

[37] P-L. Garoche. 2019. *Formal Verification of Control System Software.* Princeton.

[38] E. Geisberger and M. Broy (eds.). 2015. Living in a Networked World. Integrated Research Agenda Cyber-Physical Systems (agendaCPS). (2015), 293 pages. http://www.acatech.de/fileadmin/user_upload/Baumstruktur_nach_Website/Acatech/root/de/Publikationen/Projektberichte/acaetch_STUDIE_agendaCPS_eng_WEB.pdf.

[39] I. Gelfand and S. Fomin. 2000. *Calculus of Variations.* Dover.

[40] A. Girard, J. Agung, and G. Pappas. 2008. Approximate Simulation Relations for Hybrid Systems. *Discrete Event Dyn. Sys.* 18 (2008), 163–179.

[41] A. Girard and G. Pappas. 2007. Approximation Bisimulation Relations for Constrained Linear Systems. *Automatica* 43 (2007), 1307–1317.

[42] A. Girard and G. Pappas. 2007. Approximation Metrics for Discrete and Continuous Systems. *IEEE Trans. Autom. Control* 52 (2007), 782–798.

[43] W. Haddad and V. Chellaboina. 2008. *Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach.* Princeton University Press.

[44] E. Hairer, S. Norsett, and G. Wanner. 1993. *Solving Ordinary Differential Equations I Nonstiff Problems.* Springer.

[45] E. Hairer and G. Wanner. 1996. *Solving Ordinary Differential Equations II Stiff and Differential-Algebraic Problems.* Springer.

[46] J. He. 1994. From CSP to Hybrid Systems. In *A Classical Mind, Essays in Honour of C.A.R. Hoare*, Roscoe (Ed.). Prentice-Hall, 171–189.
[47] T. Henzinger. 1996. The Theory of Hybrid Automata. In *Proc. IEEE LICS-96*. IEEE, 278–292. Also http://mtc.epfl.ch/~tah/Publications/the_theory_of_hybrid_automata.pdf.
[48] D. Hinrichsen and A. Pritchard. 2005. *Mathematical Systems Theory I.* Springer.
[49] C. Hoare. 1969. An Axiomatic Basis for Computer Programming. *Comm. A.C.M.* 12 (1969), 576–580.
[50] HSCC. Hybrid Systems: Command and Control Conferences.
[51] ISO/IEC 13568 2002. *Information Technology – Z Formal Specification Notation – Syntax, Type System and Semantics: International Standard.* ISO/IEC 13568.
http://www.iso.org/iso/en/ittf/PubliclyAvailableStandards/c021573_ISO_IEC_13568_2002(E).zip.
[52] C.B. Jones, P. O'Hearne, and J. Woodcock. 2006. Verified Software: A Grand Challenge. *IEEE Computer* 39, 4 (2006), 93–95.
[53] C. Jones and J. Woodcock (eds.). 2008. Special Issue on the Mondex Verification. *Form. Asp. Comp.* 20 (2008), 1–139.
[54] G. Kelly. 1982. *Basic Concepts of Enriched Category Theory.* London Mathematical Society Lecture Note Series, Vol. 64, Cambridge University Press.
[55] H. Kong, F. He, X. Song, W. Hung, and M. Gu. 2002. Exponential-Condition-Based Barrier Certificate Generation for Safety Verification of Hybrid Systems. In *Proc. CAV-13*, Vol. Proc. CAV-13, Vol. 8044. Springer, LNCS, 242–257.
[56] E. Lee and S. Shesha. 2015. *Introduction to Embedded Systems: A Cyberphysical Systems Approach* (2nd. ed.). LeeShesha.org.
[57] D. Liberzon. 2012. *Calculus of Variations and Optimal Control Theory.* Princeton.
[58] J. Liu, J. Lv, Z. Quan, H. Zhao, C. Zhou, and L. Zou. 2010. A Calculus for Hybrid CSP. In *Proc. APLAS-10*, Ueda (Ed.), Vol. 6461. Springer, LNCS, 1–15.
[59] N. Lynch and F. Vaandrager. 1995. Forward and Backward Simulations Part I: Untimed Systems. *Inf. and Comp.* 121 (1995), 214–233.
[60] Mathematica. http://www.wolfram.com.
[61] A. Platzer. 2010. *Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics.* Springer.
[62] A. Platzer. 2018. *Logical Foundations of Hybrid Systems.* Springer.
[63] A. Polyanin and V. Zaitsev. 2018. *Handbook of Ordinary Differential Equations: Exact Solutions, Methods, and Problems.* C.R.C. Press.
[64] S. Prajna and A. Jadbabaie. 2004. Safety Verification of Hybrid Systems Using Barrier Certificates. In *Proc. HSCC-04*, Vol. Proc. HSCC-04, Vol. 2289. Springer, LNCS, 477–492.
[65] P. Prenter. 2008. *Splines and Variational Methods.* Dover.
[66] Retrenchment Homepage. http://www.cs.man.ac.uk/~banach/retrenchment.
[67] I. Ross. 2015. *A Primer on Pontryagin's Principle in Optimal Control.* Collegiate.
[68] H. Sagan. 1992. *Introduction to the Calculus of Variations.* Dover.
[69] R. Sanfelice. 2021. *Hybrid Feedback Control.* Princeton.
[70] S. Schneider, H. Treharne, and H. Wehrheim. 2014. The Behavioural Semantics of Event-B Refinement. *Form. Asp. Comp.* 26 (2014), 251–280.
[71] E Sekerinski and K Sere. 1998. *Program Development by Refinement: Case Studies Using the B-Method.* Springer.
[72] E. Sontag. 1998. *Mathematical Control Theory.* Springer.
[73] S. Stepney, D. Cooper, and J. Woodcock. 2000. *An Electronic Purse: Specification, Refinement and Proof.* Technical Report PRG-126. Oxford University Computing Laboratory.
[74] Symbolaris. 2014. http://www.symbolaris.org.
[75] P. Tabuada. 2009. *Verification and Control of Hybrid Systems: A Symbolic Approach.* Springer.
[76] W. Walter. 1998. *Ordinary Differential Equations.* Springer.
[77] Wikipedia. Cubic Hermite spline.
[78] Wikipedia. Duhamel's Integral.
[79] J. Woodcock. 2006. First Steps in the The Verified Software Grand Challenge. *IEEE Computer* 39, 10 (2006), 57–64.
[80] J. Woodcock and R. Banach. 2007. The Verification Grand Challenge. *JUCS* 13, 5 (2007), 661–668.
[81] J Woodcock and J Davies. 1996. *Using Z, Specification, Refinement and Proof.* Prentice Hall.
[82] N. Zhan, S. Wang, and H. Zhao. 2017. Hybrid CSP. In *Formal Verification of Simulink/Stateflow Diagrams: A Deductive Approach.* Formal Verification of Simulink/Stateflow Diagrams: A Deductive Approach, Springer, 71–90.

Received Dextobuary 2097; revised Dextobuary 2098; accepted Dextobuary 2099