Issues in Automated Urban Train Control: ‘Tackling’ the Rugby Club Problem

Richard Banach[1]

[1]School of Computer Science, University of Manchester,
Oxford Road, Manchester, M13 9PL, U.K.
richard.banach@manchester.ac.uk

Abstract. Normally, the passengers on urban rail systems remain fairly stationary, allowing for a relatively straightforward approach to controlling the dynamics of the system, based on the total rest mass of the train and passengers. However when a mischievous rugby club board an empty train and then run and jump-stop during the braking process, they can disrupt the automatic mechanisms for aligning train and platform doors. This is the rugby club problem for automated urban train control. A simple scenario of this kind is modelled in Hybrid Event-B, and sufficient conditions are derived for the prevention of the overshoot caused by the jump-stop. The challenges of making the model more realistic are discussed, and a strategy for dealing with the rugby club problem, when it cannot be prevented, is proposed.

1 Introduction

With profuse apologies to Clement Clarke Moore: *‘Twas early in the morning, when all thro’ the house, Not a creature was stirring, not even a mouse ...* aside, that is, from the stout adherents of a rugby club, who were bent on making their way to the *Métro* station, to board the otherwise empty first service of the day on the fully automated, unmanned line.[1]

As the train pulls out of the station, the dynamical variables are measured by the train system,[2] in order to gauge the weight of the passengers that have got on board — this, in order to be able to accurately predict the braking force that will be needed when the train pulls into the next station. The train becomes cognisant of the weight of the rugby club, at this point standing at the back of the train.

As the train starts to approach the next station, the rugby club start a run up the empty train towards the front. The velocity feedback control law governing the train’s travel detects a shortfall in velocity and commands additional acceleration to bring the train up to speed, thereby adding to the momentum of the whole train. The train starts to brake as it enters the next station. As the train is coming to a stop, the rugby club complete their run with a jump-stop, impulsively imparting their momentum to the train body. The train has calculated its braking force on the basis of the earlier, stationary rugby club, and has not taken into account the additional momentum. As a result of the

[1] Such as the Paris *Météor* Line 14, engineered using the B-Method.
[2] Acceleration, time taken to reach cruising speed, etc.

jump-stop, the train's braking force is inadequate, and the train overshoots its intended stopping point ... by a sufficient distance for the misalignment with the platform side doors to exceed the permitted safety margin. The only option for such excess misalignment (taking into account the demands of rush-hour throughput) is that the train does not stop but continues to the next station. Having given a cheer, the rugby club make their way to the back of the train, which still works on the basis of the original weight estimate. As the next station is approached, they start a run ... you can guess the rest. On a circular line,[3] the rugby club can amuse themselves this way all day long, with the train never stopping until the end of service. This is the Rugby Club Problem for automated urban railways.[4]

The problem of a moderate, but nevertheless unacceptable overshoot of the door position by an automated urban train is easily solved if the train doors are equidistantly spaced. Then, it is enough to have an additional door or two at the front end of the platform. The train then aims for its normal position, and if an overshoot happens, the train can carefully, but quite quickly, inch along to the next spare door position, the equidistant spacing guaranteeing that all train doors will thereby be correctly aligned.[5] But the equidistant design is not widely adopted. To have enough doors per carriage to cope with a busy rush hour in an urban environment that is populated enough to justify an urban rail solution in the first place, puts considerable demands on the structural integrity of the carriages, leading to additional costs.[6]

Putting aside levity, the aim of this paper is to demonstrate that Hybrid Event-B [8, 9] can deal fluently with the problem of modeling the kind of impulsive physics exhibited by the Rugby Club Problem. By now there are quite a number of existing case studies using Hybrid Event-B [7, 6, 2, 3, 11, 5, 4], but none of the ones published hitherto has focused on impulsive physics.

The remaining sections of the paper are as follows. In Section 2, we outline Hybrid Event-B, emphasising the elements that are useful in modelling impulsive physics. In Section 3, for lack of space, we present a very simple model of the Rugby Club Problem, displaying its essential characteristics, including how the impulsive elements are handled. In Section 4, we consider various alternatives and enhancements, which we discuss more briefly. Section 5 discusses how the Rugby Club Problem might be solved in the context of the modelling of this paper. Section 6 concludes.

2 Hybrid Event-B, and Modelling Impulsive Physics

In this section, we outline Event-B and Hybrid Event-B for a single machine. Because it is more complex, we describe Hybrid Event-B first via Fig. 1, and show how it reduces to Event-B (which of course came earlier) by erasing the more recently added elements.

Fig. 1 shows a schematic Hybrid Event-B machine. It starts with declarations of time and of a clock. Time is a first class citizen in that all variables are functions of

[3] O. K. The *Météor* line is not circular, but you get the idea.

[4] I am indebted to Thiérry Lecomte of ClearSy [14] for this delightful story [19].

[5] Such a design is visible on the Shanghai Metro's circular line 4, as well as on some other, older Shanghai Metro lines, built when train door alignment control was less precise than today.

[6] The robustness of the carriages on the Shanghai line 4 would put much heavy rail to shame.

```
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
      $clk, x, y, u :=$
        $1, x_0, y_0, u_0$
    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. 1. A schematic Hybrid Event-B machine.

time (which is read-only), explicitly or implicitly. Clocks are assumed to increase like time, but may be set during mode events. Variables are of two kinds. There are mode variables (like u) 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 typically take their values in topologically dense sets (normally $\mathbb{R}$) and which are allowed to change continuously, such change being specified via pliant events.

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

Then, the events. The *INITIALISATION* has a guard that synchronises time with the start of any run, while all other variables are assigned their initial values as usual.

Mode events are analogues of events in discrete Event-B. They can assign all machine variables (except time). The schematic *MoEv* of Fig. 1, has parameters $i?, l, o!$, (input, local, and an output), and a guard *grd*. It also has the after-value assignment specified by the before-after predicate *BApred*, which can specify the after-values of all variables (except time, inputs and locals).

Pliant events are new. They specify the continuous evolution of the pliant variables over an interval of time. Fig. 1 has a schematic pliant event *PliEv*. There are two guards: *iv*, for specifying enabling conditions on the pliant variables, clocks, and time; and *grd*, for specifying enabling conditions on the mode variables (in [8] there is a detailed discussion justifying such a design).

The body of a pliant event contains three parameters $i?, l, o!$, (input, local, and output, again) 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 contains direct assignments, e.g. of y and output $o!$ (to time dependent functions); and differential equations, e.g. specifying x via an ODE (with $\mathcal{D}$ as the time derivative).

The COMPLY clause is used to express any additional constraints that are required to hold during the pliant event via the before-during-and-after predicate *BDApred*. Typically, constraints on the permitted ranges of the pliant variables, can be placed here.

The COMPLY clause can also specify at an abstract level, e.g. stating safety properties for the event without going into detail.

Briefly, the semantics of a Hybrid Event-B 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.

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

We insist that on every subinterval $[t_i \dots t_{i+1})$ the behaviour is governed by a well posed initial value problem $\mathcal{D}\, xs = \phi(xs \dots)$ (where xs is a relevant tuple of pliant variables). Within this interval, we seek the earliest time t_{i+1} at which a mode event becomes enabled, and this time becomes the preemption point beyond which the solution to the ODE system is abandoned, and the next solution is sought after the completion of the mode event.

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

- Every enabled mode event is feasible, i.e. has an after-state, and on its completion enables a pliant event (but does not enable any mode event).[7] (1)
- Every enabled pliant event is feasible, i.e. has a time-indexed family of after-states, and EITHER: (2)

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

Thus in a well formed run mode events alternate with pliant events. The last event (if there is one) is a pliant event (whose duration may be finite or infinite). In reality, there are several semantic issues that we have glossed over in the framework just sketched. We refer to [8] for a more detailed presentation (and to [9] for the extension to multiple machines). The presentation just given is quite close to the modern formulation of hybrid systems. See e.g. [23, 22], or [13] for a perspective stretching further back.

The mode events of Hybrid Event-B, which permit the discontinuous state changes of the computational world to be represented, also allow impulsive physics to be conveniently modelled. For example, a billiard cue strikes a ball, changing its velocity

[7] 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) automatically. By this means we can ensure a mode event executes asynchronously — and if the only purpose of having an input would be to ensure this asynchronous scheduling, we can introduce the 'async' status and omit the input altogether, as in in Fig. 2.

discontinuously, or a capacitor discharges, instantaneously reducing the electrical potential across its plates to zero. However, unlike the computational world in which the programmer is at liberty to decide what discontinuous state changes take place, the physical world is governed by immutable laws, which must be adhered to to yield a useful model. Thus, in the billiard ball example, it is the conservation of momentum that determines the relationship between the physical states before and after the strike. We might say that *'Hybrid Event-B cannot do your physics for you; but it can faithfully represent the physics that you know from elsewhere.'*

Connected with the preceding is the fact that discontinuous state changes in the physical world are stimulated by forces which are 'pure impulses'. And whereas discontinuous change can be represented quite directly in Hybrid Event-B, these pure impulses cannot. Physicists and engineers speak of such impulses as 'delta functions' — 'zero everywhere except at a single point, were they are infinite, but with a finite integral'. Mathematically, that last phrase is meaningless; delta functions are not functions, but so-called distributions [25, 24, 18]. The nearest we get in Hybrid Event-B (or any other similar formalism) to the representation of a pure impulse is the (syntactic) description of the mode event that encapsulates the discontinuous change that results from the impulse. The occurrence of the mode event (at runtime) corresponds to the occurrence of the physical impulse causing the discontinuous change of state.

3 A Simple Rugby Club Problem Scenario

In this section, we present a very simple model of the rugby club scenario, formalised in Hybrid Event-B and, in particular, utilising the insights about impulsive physics just discussed. The model itself is in Fig. 2.

The model depends on a number of constants (which would be declared in a CONTEXT, which we do not show). There are: the phases of the model stored in the *mode* variable: *STAT*ionary; *ACCEL*erating; *CRUISE*ing; *DECEL*erating. There is also: *BIGT*, an intial value for a clock that is bigger than any value that could trigger the enabledness of any mode event; M_T, the mass of the train; M_{rc}, the mass of the rugby club; V_{cr} the cruising speed of the train; V_{rcr} the rugby club's running speed relative to the train's speed (when the members of the rugby club are, in fact, running).

A number of variables contain the state of the model. There is *mode*, already mentioned. There are a number of variables representing mass: m_{in}, the inertial mass of the system at any time; m_{pcv}, the mass perceived by the train at any time (based on the dynamical properties that it measures and the moments in time that it does so); v_{rcr}, the rugby club's running speed relative to the train at any time (regardless of whether the rugby club are, in fact, running or not at that time); *brTime*, the train's concept of the needed duration for the braking period, at the start of the braking period. These variables are mode variables, because they only need to get updated via mode events.

There are also pliant variables: m_{eff}, the effective mass of the system, i.e. the point mass which, when traveling at the train's velocity, would possess the same momentum as the whole train plus rugby club system, thus offering the same resistance to change in momentum as the whole system — it changes continuously when the rugby club is running during acceleration or braking, due to the continuously changing relative

```
MACHINE $RugbyClub\_1$
CLOCK $clk\_A$
VARIABLES $mode, m_{in}, m_{pcv}, v_{rcr}, brTime$
PLIANT $m_{\mathrm{eff}}, v_T, brDist$
INVARIANTS
  $mode \in \{STAT, ACCEL, CRUISE, DECEL\}$
  $m_{in}, m_{pcv}, v_{rcr}, brTime \in \mathbb{R}^+, \mathbb{R}^+, \mathbb{R}^+, \mathbb{R}^+$
  $m_{\mathrm{eff}}, v_T, brDist \in \mathbb{R}, \mathbb{R}, \mathbb{R}$
EVENTS
  INITIALISATION
    BEGIN
      $clk\_A := BIGT$
      $mode := STAT$
      $m_{in}, m_{pcv} := M_T, M_T$
      $v_{rcr} := 0$
      $m_{\mathrm{eff}} := M_T$
      $v_T := 0$
      $brDist := 0$
      $brTime := 0$
    END
  TrainStationary                                  pliant
    WHEN $mode = STAT$
    THEN
    COMPLY $\mathsf{CONST}(m_{\mathrm{eff}}, v_T, brDist)$
    END
  RugbyClubBoards                                  async
    WHEN $mode = STAT$
    THEN
      $m_{in} := M_T + M_{rc}$
      $m_{\mathrm{eff}} := M_T + M_{rc}$
    END
  TrainStarts                                      async
    BEGIN
      $mode := ACCEL$
      $clk\_A := 0$
    END
  TrainAccelerating                                pliant
    WHEN $mode = ACCEL$
    COMPLY $\mathsf{CONST}(m_{\mathrm{eff}})$
    THEN
      $\mathcal{D}\, v_T = F_A / m_{in}$
      $brDist := 0$
    END
  TrainAtSpeed
    WHEN $mode = ACCEL \wedge v_T = V_{cr}$
    THEN
      $mode := CRUISE$
      $m_{pcv} := (F_A\, clk\_A)/V_{cr}$        // $\ldots = m_{in}$
      $m_{\mathrm{eff}} := m_{in}$
    END
  TrainCruising                                    pliant
    WHEN $mode = CRUISE$
    THEN
      $\mathcal{D}\, v_T = -K(v_T - V_{cr})$
      $brDist := 0$
      $m_{\mathrm{eff}} := (M_T + M_{rc}) + \dfrac{M_{rc} v_{rcr}}{v_T}$
    END
...   ...
```

```
...   ...
  RugbyClubStartsRun                               async
    WHEN $mode = CRUISE$
    THEN
      $v_{rcr} := V_{rcr}$
      $v_T := V_{cr} - \dfrac{M_{rc}\, V_{rcr}}{M_T + M_{rc}}$
      $m_{\mathrm{eff}} := \dfrac{(M_T + M_{rc})}{\left[1 - \dfrac{V_{rcr}}{V_{cr}}\left[1 + \dfrac{M_T}{M_{rc}}\right]^{-1}\right]}$
    END
  TrainBrakes                                      async
    WHEN $mode = CRUISE$
    INIT $|\, v_T - V_{cr} \,| < \epsilon$
    THEN
      $mode := DECEL$
      $brDist := -\dfrac{m_{pcv}}{F_D} \dfrac{V_{cr}^2}{2}$
      $brTime := \dfrac{m_{pcv}\, V_{cr}}{F_D}$
    END
  TrainDecelerating                                pliant
    WHEN $mode = DECEL$
    THEN
      $\mathcal{D}\, v_T = -\dfrac{F_D}{M_T + M_{rc}}$
      $\mathcal{D}\, brDist = v_T$
      $m_{\mathrm{eff}} := (M_T + M_{rc}) + \dfrac{M_{rc} v_{rcr}}{v_T}$
    END
  RugbyClubJumpStop                                async
    WHEN $mode = DECEL \wedge v_T > 0$
    THEN
      $v_{rcr} := 0$
      $v_T := \dfrac{m_{\mathrm{eff}}}{m_{in}} v_T$
      $m_{\mathrm{eff}} := m_{pcv}$
    END
  TrainStopSucceed
    WHEN $mode = DECEL \wedge v_T = 0 \wedge$
      $|\, brDist \,| \leq BRTOL$
    THEN
      $clk\_A := BIGT$
      $mode := STAT$
      $m_{in}, m_{pcv} := M_T, M_T$
      $brDist := 0$
      $brTime := 0$
    END
  TrainStopFail
    WHEN $mode = DECEL \wedge v_T = 0 \wedge$
      $|\, brDist \,| > BRTOL$
    THEN
      $clk\_A := 0$
      $mode := ACCEL$
      $m_{in}, m_{pcv} := M_T, M_T$
      $brDist := 0$
      $brTime := 0$
    END
END
```

Fig. 2. A simple Hybrid Event-B model of the urban rail Rugby Club Problem.

proportions that the train and the rugby club contribute to the overall momentum during the accelerating or braking episodes; v_T, the speed of the train at any time; *brDist*, the current remaining distance during the braking period until the train comes to a standstill, as computed by the train according to the the dynamical properties that it measures.

In reality, not all of these variables are strictly necessary. Many can be dispensed with as they can be re-expressed in terms of constants and other variables. The variables in this category are: m_{in}, m_{pcv}, v_{rcr}, m_{eff} and *brTime*. We nevertheless retain them in order to make the ensuing explanation of the model easier to follow.

The invariants are trivial typing invariants in this simple model: *mode* is as described earlier, and the others are all either reals or non-negative reals. We discuss some possibilties for more complex invariants later.

We turn to what the model actually does. In order to save space in Fig. 2, we have economised on some notational matters. Thus: We have decanted events' STATUS declarations to a decoration at the end of the line containing the event name (where the STATUS is not 'ordinary'). We have used the 'async' STATUS to ensure a mode event does not execute eagerly.[8] We have slightly generalised the CONST declaration of [2] to cover a list of (pliant) variables that are to stay constant during the execution of a pliant event.

INITIALISATION starts the model with the train stationary in a station with no one on board. A clock *clk_A*, is set to an innocuous value *BIGT*; the *mode* is *STAT*; all the masses are set to be the train's inertial mass M_T; the train's velocity and the rugby club's relative velocity are set to zero; and all other variables are of no interest and are also set to zero.

The ensuing pliant event *TrainStationary* just perpetuates this state of affairs — all mode variables cannot change, and the pliant variables are held constant via the CONST declaration.

At some point during this phase the async event *RugbyClubBoards* is executed. Although boarding clearly does not take place instantaneously, only the overall change in mass makes any difference, and so there is no harm in modelling the process as an impulsive change to the mass during this event. The inertial mass m_{in} becomes $M_T{+}M_{rc}$, as does the effective mass m_{eff} (since the train system behaves as a single mass at this stage). Everything else stays the same. In particular, the train's perceived mass m_{pcv} remains unchanged since the train is, as yet, unaware of the rugby club. After this the *TrainStationary* event resumes, all variables maintaining their values, whether old values or newly acquired values.[9]

At some point after this the dynamics starts, and for this, we assume an conventionally idealised setup. Thus we assume the track is straight and level, the movement of the train is frictionless and suffers no other impediment (such as air resistance), and the train can be treated as one (or several) point mass(es) for the purpose of dynamical calculations.

[8] Thus, 'STATUS async' is an abbreviation for the semantic device of giving the mode event an external input which is not used in the event's body. See footnote 7.

[9] In fact, *RugbyClubBoards* remains enabled during the resumed *TrainStationary* event, so could execute again. But *RugbyClubBoards* is async and idempotent, so no harm would be done.

In complex situations, dynamics is best treated via the d'Alembert-Lagrange approach, or an equivalent. See e.g. [17, 15]. The foundations are not in fact as uncomplicated as the ancient pedigree of this subject might suggest; [12] gives a good discussion, not to mention the gargantuan [21]. For us, it will suffice to stick to a fairly low-level approach, *provided* we remember that Newton's Second Law equates *force* to rate of change of *momentum*, and not to mass times acceleration, as is usually stated, and to which the more accurate form usually reduces.

So, async event *TrainStarts* executes. It changes the mode to *ACCEL*erating and starts the clock. It thus enables the *TrainAccelerating* pliant event which states how the pliant variables change. Since the rugby club are stationary, the effective mass m_{eff} remains CONSTant at its value at the start of *TrainAccelerating*. The braking distance variable *brDist* is not needed yet, so is kept at zero.[10] The nontrivial element of the *TrainAccelerating* event is the ODE that equates the rate of change of the train's momentum $\mathcal{D}\,(m_{in}\, v_T)$ to the force applied by the train. The latter is assumed to be a constant accelerating force F_A. Since there is no relative motion between the train and rugby club, and all the train and rugby club mass is treated as concentrated at the centre of gravity, we can take the mass element to be the intertial mass m_{in}, and we derive the statement found in Fig. 2.

Acceleration continues until the train achieves cruising velocity, detected by the guard $v_T = V_{cr}$ of the mode event *TrainAtSpeed*. This turns off the accelerating force and changes the mode to *CRUISE*ing. This also enables the train to calculate its overall perceived mass m_{pcv} from the information it has, namely clock value clk_A, applied force F_A and cruise velocity V_{cr}. Of course, since the motion has been so simple thus far, a straightforward application of Newtonian mechanics (namely, that change of momentum $m_{pcv} \times V_{cr}$ equals duration of applied force $F_A \times clk\_A$) shows that the answer m_{pcv}, is m_{in}, as noted in the accompanying comment, but the train can only use the information available to it, so we show the assignment to m_{pcv} expressed using those quantities.

TrainAtSpeed enables the *TrainCruising* pliant event. *brDist* is still not needed, so is assigned as previously. The train velocity v_T is controlled by a linear constant coefficients ODE, impelling v_T towards the stable equilibrium value V_{cr}. Since $v_T = V_{cr}$ immediately after *TrainAtSpeed*, there is no change in velocity at this time. Similarly, the effective mass m_{eff} remains as before, which is easy to see in the direct assignment to m_{eff} when we notice that $v_{rcr} = 0$ during this period.

During *TrainCruising*, async mode event *RugbyClubStartsRun* is enabled, and at some point is executed. Now the dynamics gets more interesting. Again we idealise the change of state as an impulsive change, since only the overall change in momentum matters, and the dynamics is completely lossless. The rugby club's relative velocity with

[10] This could also have been handled via a CONST declaration. In fact, that would have been more convenient, since assignment to a (time dependent, in general) expression generates a verification condition to check that the initial value of the expression agrees with the value on entry to the pliant event, in order to ensure right continuity of the variable's history at the entry point to the pliant event, as required by the semantics [8]. Not mentioning *brDist* at all would entail the default behaviour for pliant variables during pliant events, namely of constraining them to simply obey any relevant invariants. This would be inappropriate here.

respect to the train v_{rcr}, becomes V_{rcr}. Since momentum is conserved, using primes for after-values, we can write:

$$(M_T + M_{rc})v_T = (M_T + M_{rc})v'_T + M_{rc}V_{rcr} \tag{3}$$

from which, noting that $v_T = V_{cr}$, we derive the assignment to v_T that we see in *RugbyClubStartsRun*. The train effective mass m_{eff} becomes the mass that is needed to generate the momentum on either side of (3) when the velocity is the new train velocity. A slightly longer calculation, equating (3) to $m'_{\text{eff}}v'_T$, is needed to derive the expression for m_{eff} (given in terms of the cruise velocity V_{cr}) that appears in *RugbyClubStartsRun*.

After *RugbyClubStartsRun*, *TrainCruising* is still enabled. Since the train velocity is no longer V_{cr}, the feedback control law in *TrainCruising* now has work to do. Implicitly, an accelerating force is applied to impel the train velocity v_T towards V_{cr}, and it does work that adds to the total momentum of the system. Note that as v_{rcr} is non-zero, having become V_{rcr}, and given that v_T changes, so does m_{eff}, as can be derived from (3), reflecting the changing proportion of the overall momentum that the rugby club's relative run velocity contributes.

When v_T has returned close enough to V_{cr}, the async mode event *TrainBrakes* becomes enabled — we are assuming that the train velocity has recovered before the train starts to brake. We assume the train knows where it is relative to the next stopping position, and initiates braking at a point where, according to the train's perception about its dynamics, applying its fixed braking force F_D for an appropriate time will bring it to a halt just where needed. We assume that the train still imagines its overall mass is the originally calculated m_{pcv}, and taking the velocity to be V_{cr}, a simple Newtonian mechanics calculation of the (quadratic) displacement generated by a constant force yields the *brDist* value assigned in *TrainBrakes*, assuming further that the next stopping position is the origin of distance measurements, and that positive distances are oriented beyond the stopping position. The time taken to come to a halt is recorded in *brTime* — it is just the time needed to consume all of the assumed momentum $m_{pcv}V_{cr}$ by applying a force of magnitude F_D.

TrainBrakes changes the mode to *DECEL*erating, and thus enables pliant behaviour *TrainDecelerating*. During this period, it is the laws of physics, and not the train's perceptions, that determine what happens. Thus, the rate of change of velocity is governed by the momentum form of Newton's Law:

$$\mathcal{D}\left((M_T + M_{rc})v_T + M_{rc}V_{rcr}\right) = -F_D \tag{4}$$

In (4), only v_T can vary, the other symbols being constants. Thus we derive the ODE for v_T in *TrainDecelerating*. And v_T gives the time derivative of *brDist*. The effective mass m_{eff} is given by the same formula as in *TrainCruising*, for exactly the same reasons.

At some point during *TrainDecelerating*, but while the velocity is still positive, the rugby club come to the end of their run. The momentum that they 'stole' from the train when they initiated their run, and which was unknowingly made up by the feedback control law during *TrainCruising*, is now suddenly dumped back into the train when they do their jump-stop.

The physical consequences of this process are described in the async mode event *RugbyClubJumpStop*. The rugby club relative run velocity v_{rcr} changes from V_{rcr} to zero. Since the train system now behaves once more like a point mass, the effective mass must likewise become m_{in}. The process is governed by conservation of momentum, which, using primes for after-values as usual, yields the following:

$$m_{\mathrm{eff}}\, v_T = m_{in} v'_T = m_{pcv} v'_T = m'_{\mathrm{eff}} v'_T \tag{5}$$

This explains the assignments to the variables in *RugbyClubJumpStop*.

Once *RugbyClubJumpStop* has executed, *TrainDecelerating* is enabled once more.[11] But now, the train velocity which the decelerating phase has to deal with is suddenly greater than before. So the braking phase is extended compared with its previously anticipated duration.

It is intuitively clear that if the rugby club consists of extreme lightweights, and that if they run extremely slowly, the effect on the braking episode will be small due to the small amount of momentum at issue. Equally, if the rugby club are all very heavy, and they run very fast, then the effect on the braking episode will be more appreciable.

Mode events *TrainStopSucceed* and *TrainStopFail* handle these two possibilities. One or other is triggered when v_T hits zero (and the mode is still *DECEL*erating). The ideal stopping position is at $brDist = 0$. So if the discrepancy between the ideal and actual stopping positions when v_T hits zero does not exceed *BR*aking*TOL*erance, then *TrainStopSucceed* executes, and the train stops at the station, as it should. The state returns to its initial configuration and the rugby club can alight (presumably disappointed). The whole scenario can then repeat.

However, if the discrepancy between the ideal and actual stopping positions when v_T hits zero exceeds *BR*aking*TOL*erance, then the train returns to *ACCEL*erating mode, and the train moves towards the next station, with the (presumably elated) rugby club on board. In this case we have the classic rugby club problem which can then also repeat.

3.1 Analysis of the Jump-Stop Phenomenon

In this section we analyse more precisely the distinction between the *TrainStopSucceed* and *TrainStopFail* cases.

During an execution of *TrainDecelerating*, if the intial velocity of the train is v_{IN} and the pliant event executes for a time t_{EX}, then after this period, the velocity and distance travelled become:

$$v_{IN} - \frac{F_D\, t_{EX}}{(M_T + M_{rc})} \qquad \text{and} \qquad v_{IN}\, t_{EX} - \frac{F_D\, t_{EX}^2}{2(M_T + M_{rc})} \tag{6}$$

respectively. To work out the implications of the jump-stop, we need to consider two such episodes, separated by a *RugbyClubJumpStop*.

The braking period starts with the train moving forward with velocity V_{cr}. Suppose the rugby club do their jump-stop t_{JS} later than the start of braking. Then, substituting

[11] While *TrainDecelerating* executes, *RugbyClubJumpStop* is (more or less) always enabled, but as in other cases, it is an async event and its effect is idempotent, so executing it again would have no discernible effect.

into (6), after the first braking episode, the velocity and distance travelled become:

$$v_{JS} = V_{cr} - \frac{F_D\, t_{JS}}{(M_T + M_{rc})} \qquad \text{and} \qquad d_{JS} = V_{cr}\, t_{JS} - \frac{F_D\, t_{JS}^2}{2(M_T + M_{rc})} \tag{7}$$

Then comes the jump-stop. According to *RugbyClubJumpStop*, the velocity needs to be rescaled by m_{eff}/m_{in}, which increases the velocity expression in (7) to:

$$v_{JS} \left[m_{in} + \frac{M_{rc}\, V_{rcr}}{v_{JS}} \right] \Big/ \, m_{in} \; = \; V_{cr} - \frac{F_D\, t_{JS}}{(M_T + M_{rc})} + V_{rcr} \left[1 + \frac{M_T}{M_{rc}} \right]^{-1} \; = \; v'_{JS} \tag{8}$$

Braking is then completed by another *TrainDecelerating* episode. This time the initial velocity is v'_{JS}. Using (6) with this intial value, the pliant behaviour executes until the velocity drops to zero. Naming this duration t_{HALT}, it is given by:

$$v'_{JS} - \frac{F_D\, t_{HALT}}{(M_T + M_{rc})} \; = \; 0 \qquad \text{thus} \qquad t_{HALT} \; = \; \frac{(M_T + M_{rc})\, v'_{JS}}{F_D} \tag{9}$$

and therefore, the distance covered in the second *TrainDecelerating* episode is, by (6):

$$d_{HALT} \; = \; v'_{JS}\, t_{HALT} - \frac{F_D\, t_{HALT}^2}{2(M_T + M_{rc})} \tag{10}$$

The total distance travelled during braking is therefore $d_{TOT} = d_{JS} + d_{HALT}$, subject to the constraint that $v_{JS} > 0$. If we call $brDist_{TOT}$, the value of *brDist* assigned by *TrainBrakes*, it is the discrepancy between d_{TOT} and $brDist_{TOT}$ that must be compared to *BRTOL* to determine whether *TrainStopSucceed* or *TrainStopFail* will be enabled. We find:

$$\begin{aligned}
d_{TOT} \; &= \; V_{cr}\, t_{JS} - \frac{F_D\, t_{JS}^2}{2(M_T + M_{rc})} + v'_{JS}\, t_{HALT} - \frac{F_D\, t_{HALT}^2}{2(M_T + M_{rc})} \\
&= \; V_{cr}\, t_{JS} - \frac{F_D\, t_{JS}^2}{2(M_T + M_{rc})} + \frac{(M_T + M_{rc})\, v'^2_{JS}}{F_D} - \frac{F_D\, (\frac{(M_T + M_{rc})\, v'_{JS}}{F_D})^2}{2(M_T + M_{rc})} \\
&= \; V_{cr}\, t_{JS} - \frac{F_D\, t_{JS}^2}{2(M_T + M_{rc})} + \frac{(M_T + M_{rc})\, v'^2_{JS}}{2\, F_D} \\
&= \; V_{cr}\, t_{JS} - \frac{F_D\, t_{JS}^2}{2(M_T + M_{rc})} \\
&\quad + \; \frac{(M_T + M_{rc})}{2\, F_D} \left(V_{cr} - \frac{F_D\, t_{JS}}{(M_T + M_{rc})} + V_{rcr} \left[1 + \frac{M_T}{M_{rc}} \right]^{-1} \right)^2
\end{aligned} \tag{11}$$

After a bit more working we get:

$$\begin{aligned}
d_{TOT} = \; \frac{(M_T + M_{rc})}{2\, F_D} \Bigg[& V_{cr}^2 + V_{rcr}^2 \left[1 + \frac{M_T}{M_{rc}} \right]^{-2} + 2\, V_{rcr} V_{cr} \left[1 + \frac{M_T}{M_{rc}} \right]^{-1} \\
& - 2\, V_{rcr} \frac{F_D\, t_{JS}}{(M_T + M_{rc})} \left[1 + \frac{M_T}{M_{rc}} \right]^{-1} \Bigg]
\end{aligned} \tag{12}$$

So

$$d_{TOT} < \frac{(M_T + M_{rc})}{2\,F_D}\left[V_{cr}^2 + V_{rcr}^2\left[1+\frac{M_T}{M_{rc}}\right]^{-2} + 2\,V_{rcr}V_{cr}\left[1+\frac{M_T}{M_{rc}}\right]^{-1}\right] \qquad (13)$$

The last step follows, because in the last two terms of (12), the negative one cannot exceed the positive one in magnitude because of the constraint on t_{JS} coming from $v_{JS} > 0$, which implies that t_{JS} reaches its maximum when $v_{JS} = 0$, at which point these last two terms cancel.

Therefore, if we can arrange it that $|\,brDist_{TOT} + d_{TOT}\,| < BRTOL$, we will always execute *TrainStopSucceed* at the end of the braking process, and will never execute *TrainStopFail*. We will have surmounted the rugby club problem.

We can express this insight as an additional, nontrivial invariant, where *HYP* denotes the relationships between the various constants of the model that have to be true in order that $|\,brDist_{TOT} + d_{TOT}\,| < BRTOL$ holds:

$$HYP \;\vdash\; mode = DECEL \wedge v_T = 0 \;\Rightarrow\; |\,brDist\,| \leq BRTOL \qquad (14)$$

Thus, the enabledness of *TrainStopSucceed* at the crucial moment becomes provable.

Regarding $brDist_{TOT}+d_{TOT}$, which equals the last two terms of (13), we note that the V_{rcr}^2 term will be negligible in magnitude compared with the $V_{rcr}V_{cr}$ term. This enables us to derive a simple criterion that will be adequate for most engineering purposes:

$$brDist_{TOT} + d_{TOT} \;<\; \frac{V_{rcr}\,V_{cr}\,M_{rc}}{F_D} \qquad (15)$$

4 The Rugby Club Problem — Further Discussion

The details of the control strategy actually used for urban rail control are commercially confidential, for obvious reasons. Nevertheless, it seems clear that the fact that there is a rugby club problem at all, signals a likely cause of it as being the discrepancy between a control strategy based on pure kinematics and one based on the complete dynamics. In this section, we briefly some discuss issues for more realistic modelling.

Several factors would need to be taken into account in a more realistic model: the track will not be straight and level; it will not sustain frictionless train travel; the train's wheels will not always make perfect rolling contact with the track (there will be some skidding at times); the control laws will not be as simple as we have chosen them to be in our models; in the confines of an underground tunnel, air resistance will cause significant drag on the train. And so on.

All these things will soak up some of the momentum of the train as it travels, requiring work from the engine to maintain speed. Simple realistic models of these phenomena will not be available. The best one might hope for, would be phenomenological models that predicted the relevant losses, based on tabulated data taken over many journeys under a variety of conditions. These data would have to be specific to each section of the route, and dealing with these aspects could seriously complicate the design of the critical code controlling the train's motion.

5 'Tackling' the Rugby Club Problem

Above, we suggested that if appropriate relationships could be made to hold between the various constants that characterised our model, then the rugby club problem might be overcome. In this section, we discuss how the rugby club problem may be addressed when such choices of constants are not available for whatever reason, while remaining within the simple modelling framework of Fig. 2.

In our model, the principal cause of the loss of coherence between the train's view of the dynamics and the physical reality could be attributed to the fact that the control law for the cruise phase was based exclusively on the train's velocity, whereas the true physics of the situation requires the accounting of momentum.

The obvious suggestion then, would be to change the control laws for the various phases of the dynamics to account for momentum more accurately. In our extremely idealised models this would not be hard to do, because in such simple models, the relationships between velocity and momentum are straightforward, and the cruise phase could easily detect how much momentum it had given away as it brought the train back up to speed. The train could then approach the stopping point more cautiously, knowing that the momentum it had given away would have to be given back soon.

However, when we consider doing the same thing in the context of the more realistic models contemplated in Section 4, this is easier said than done. The rugby club steals momentum from the train, but so do all the other sources of non-ideal motion that we mentioned. Distinguishing between 'natural losses' and 'unnatural losses' becomes nontrivial. Nevertheless, if natural and unnatural could be distinguished clearly enough, an optional 'more cautious stopping strategy' offers a potential way forward.

6 Summary and Conclusions

In the preceding sections we outlined the essentials of Hybrid Event-B, with a special focus on how impulsive physics can be handled. Then we constructed a Hybrid Event-B model of the rather engaging rugby club problem scenario described in the Introduction. For the purposes of arriving at a reasonably clear exploration of the rugby club problem which nevertheless fitted in a fairly short paper, our model had to simplify and idealise the situation rather severely. It was thus suffused with point mass and lossless dynamics in the familiar style of classical mechanics. The precision of the model allowed us to derive conditions that distinguished between the non-disruptive and disruptive case of the rugby club dynamics, and we discussed some options for adding more complex invariants to the model, based on these. We then discussed possibilities for reducing the degree of idealisation in the model, and thus the prospects for making it more realistic, thereby bringing it closer to applicability in practice.

It is worthwhile, at this point, making an observation about how the stated provability of the additional invariants that were mentioned came about. Most of the analysis of this paper was performed in a fairly *ad hoc* manner. When dealing with a situation described by physical theories, this is, more or less, unavoidable. It follows in turn because physical theories are almost always expressed using a family of equalities. As such, any of the participating variables may (in the given situation) carry input values, with the other variables acquiring their values from the demanded equalities, as outputs. So the derivation process is not structured in a manner that is fixed at the outset, in the way that formal development processes tend to be. However, once the *ad hoc* reasoning has yielded its fruits, we can take a step back, and restructure what has been discovered in a manner that better fits a formal development process. It is in this manner that the provability that is claimed of the additional invariants emerges.

In the last section, we addressed how this modelling exercise could be used to overcome the rugby club problem, in cases where it could not be prevented by choosing appropriate constants. The crux of the matter would be to centre the control system for the train more firmly on the momentum dynamics of the physical system, than on purely kinematic aspects. Confidence in this assertion is supported by the fact that although a rugby club may be able to outwit a train control system whose design is insufficiently suspicious, they cannot cheat the laws of physics.

It is instructive to note the very major role played by physics knowledge in the exercise undertaken in this paper. Although computer scientists often find it convenient to downplay or neglect the influences of non-computing disciplines in the design of cyberphysical systems [16, 13] (see, for example, the balance of content in references such as [20, 1]), the importance of such influences cannot be denied, and the present exercise shows this eloquently. Cyberphysical systems are truly multidisciplinary and it is unwise to neglect any of the disciplines that contribute to a given system while emphasising just one (e.g. just the computing viewpoint). See [10] for a review of some of the less obvious elements that impact cyberphysical systems, discussed from a mathematical viewpoint.

References

1. Alur, R.: Principles of Cyberphysical Systems. MIT Press (2015)
2. Banach, R.: Pliant Modalities in Hybrid Event-B. In: Liu, Woodcock, Zhu (eds.) Proc. Jifeng He Festschrift 2013. LNCS, vol. 8051, pp. 37–53. Springer (2013)
3. Banach, R.: The Landing Gear System in Multi-Machine Hybrid Event-B. Int. J. Soft. Tools for Tech. Trans. (2015), to appear
4. Banach, R.: Formal Refinement and Partitioning of a Fuel Pump System for Small Aircraft in Hybrid Event-B. In: Bonsangue, Deng (eds.) Proc. IEEE TASE-16. pp. 65–72. IEEE (2016)
5. Banach, R.: Hemodialysis Machine in Hybrid Event-B. In: Butler, Schewe, Mashkoor, Biro (eds.) Proc. ABZ-16. vol. 9675, pp. 376–393. Springer, LNCS (2016)
6. Banach, R., Butler, M.: A Hybrid Event-B Study of Lane Centering. In: Aiguier, Boulanger, Krob, Marchal (eds.) Proc. CSDM-13. pp. 97–111. Springer (2013)
7. Banach, R., Butler, M.: Cruise Control in Hybrid Event-B. In: Liu, Woodcock, Zhu (eds.) Proc. ICTAC-13. LNCS, vol. 8049, pp. 76–93. Springer (2013)
8. Banach, R., Butler, M., Qin, S., Verma, N., Zhu, H.: Core Hybrid Event-B I: Single Hybrid Event-B Machines. Sci. Comp. Prog. 105, 92–123 (2015)
9. Banach, R., Butler, M., Qin, S., Zhu, H.: Core Hybrid Event-B II: Multiple Cooperating Hybrid Event-B Machines. Sci. Comp. Prog. 139, 1–35 (2017)
10. Banach, R., Su, W.: Cyberphysical Systems: A Behind-the-Scenes Foundational View. In: Mashkoor, Thalheim, Wang (eds.) Proc. Klaus-Dieter Schewe Festschrift 2018. College Publications (2018), to appear.

11. Banach, R., Van Schaik, P., Verhulst, E.: Simulation and Formal Modelling of Yaw Control in a Drive-by-Wire Application. In: Proc. FedCSIS IWCPS-15. pp. 731–742 (2015)
12. Bloch, A., Krishnaprasad, P., Murray, R., Baillieul, J., Crouch, P., Marsden, J., Zenkov, D.: Nonholonomic Mechanics and Control. Springer (2015)
13. Carloni, L., Passerone, R., Pinto, A., Sangiovanni-Vincentelli, A.: Languages and Tools for Hybrid Systems Design. Foundations and Trends in Electronic Design Automation 1, 1–193 (2006)
14. ClearSy: http://www.clearsy.com/
15. Fasano, A., Marmi, S.: Analytical Mechanics. Oxford University Press (2013)
16. Geisberger, E., Broy (eds.), M.: Living in a Networked World. Integrated Research Agenda Cyber-Physical Systems (agendaCPS) (2015), http://www.acatech.de/fileadmin/user_upload/Baumstruktur_nach_Website/Acatech/root/de/Publikationen/Projektberichte/acaetch_STUDIE_agendaCPS_eng_WEB.pdf
17. Goldstein, H., Poole, C., Safko, J.: Classical Mechanics. Addison Wesley (2001)
18. Horvarth, J.: Topological Vector Spaces and Distributions. Dover (2012)
19. Lecomte, T.: Atelier B has Turned 20. In: Proc. ABZ-16. vol. 9675, p. XVI. Springer, LNCS (2016)
20. Lee, E., Shesha, S.: Introduction to Embedded Systems: A Cyberphysical Systems Approach. LeeShesha.org, 2nd. edn. (2015)
21. Papastavridis, J.: Analytical Mechanics: A Comprehensive Treatise on the Dynamics of Constrained Systems. World Scientific, 2nd. edn. (2014)
22. Platzer, A.: Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics. Springer (2010)
23. Tabuada, P.: Verification and Control of Hybrid Systems: A Symbolic Approach. Springer (2009)
24. Treves, F.: Topological Vector Spaces, Distributions and Kernels. Dover (2007)
25. Zemanian, A.: Distribution Theory and Transform Analysis: An Introduction to Generalized Functions, with Applications. Dover (2003)