A new design methodology for public transport stations of a multimodal hub

F. Kaakai[1], S. Hayat[2] & A. El Moudni[1]
[1]*Laboratoire Systèmes et Transports (S.e.T),*
Université de Technologie de Belfort-Montbéliard (UTBM), France
[2]*Institut National de Recherche sur les Transports et leur Sécurité (INRETS), France*

Abstract

The purpose of this paper is to present *a new design methodology* which allows one to compute more precisely, and thus at a lower cost, the station platform surfaces able to absorb the peaks of travellers during rush periods.

This new design methodology is based on a *model* of "public transport stations" which can be considered as a hybrid dynamic system. This model, based on Synchronized Hybrid Petri Nets, describes the functioning and the behaviour of public transport stations through a set of non-linear and piecewise continuous equations. This non-linear hybrid formalism is complex. That is why we present in this paper an algorithmic method which simplifies this theoretical formalism and makes it exploitable for design applications.

This new design methodology applied to a rush-hour period scenario and compared with a rough design procedure shows that the space saving can be higher than 50% for only one station.

Keywords: multimodal hub, public transport, mode, connections, stations, hybrid, Petri-nets, model, platform surface, analytical design algorithm.

1   Introduction

Some standardisation works dedicated to the *service quality in public transports* like the standard EN 13816 published in 2002 by the European Committee for Standardization (ECS [8]) specify that the maximal density of travellers on station platforms must not outnumber 6 persons per m². Nevertheless, we can observe that some public stations, especially into busy multimodal hubs, do not

respect this limit during rush hours. When the travellers' concentration becomes much too high in public stations – for instance, 670 000 travellers/day use one of the biggest multimodal hubs of Paris named Chatelet-Les-Halles in order to practice intermodality (Luquet [4]) - the most fragile persons (children, old people, etc.) can be seriously affected by the movements of the crowd when the subway or the train arrives at the station: *faintness, scuffles, falls, trampling,* etc. Anyway, these dangerous situations deteriorate the service quality of public transport and make them less attractive.

In order to respect the service quality standard, a solution consists in designing platforms so that they can absorb these peaks of travellers. However, the method used, in practice, to design platforms is too rough: it is usual to calculate an average number of travellers (from statistical data), to consider a maximal traveller density and then to introduce a safety factor. This rough procedure may lead to platform surfaces too small or too big, which are incompatible with the economic and technical constraints of public transport managers.

To design platforms more precisely, it is necessary to understand well, thanks to an appropriate model, the *internal functioning* of a public transport station and the *behaviour of this system* when peaks of travellers are applied as input data. Then, this model will be analysed and exploited through simulation scenarios in order to assess the maximal concentration of travellers into the station and then to calculate its appropriate platform surface.

2 Systemic definition of public transport stations

Public transport stations consist of (i) a platform characterized by a surface (S) and a maximal density of travellers (d_{max}), (ii) the Public Transport (with a maximum load $C_{max,k}$) when it is parked at the platform level, and (iii) the travellers into the public transport vehicles and the travellers on the platform (see Fig. 1).
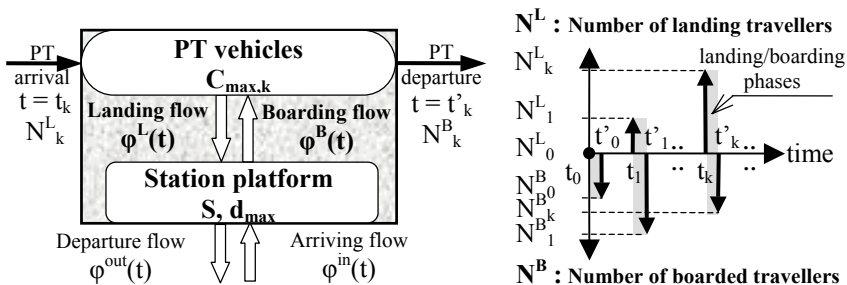

Figure 1:   Systemic definition of public transport stations.

The instances noted t_k corresponding to the public transport arrivals at the platform form a set of external events noted ExEv_arr = {t_1,···, t_k,···} whereas the instances noted t'_k corresponding to their departures from the station form a set

of external events noted ExEv_dep={t'$_1$,···, t'$_k$,···}. Thus, the landing and boarding phases take place during the interval [t$_k$ t'$_k$[(see Fig. 1):

− **at t$_k$:** N^L_k travellers inside the public transport are going to land on the platform at a certain flow of travellers per time unit noted φ^L_k(t);

− **at t'$_k$:** N^B_k travellers have boarded onto the public transport at a certain flow of travellers per time unit noted φ^B_k(t).

Once the travellers land on the platform, they leave the station at a certain flow φ^{out}(t) in order to take another transportation mode (intermodality) or to leave the multimodal hub. At the same time, travellers arrive at the station (on the platform) at a certain flow noted φ^{in}(t) in order to board onto the public transport.

3 Hybrid Petri net model of public transport stations

The successive arrivals of public transport at their respective station into the multimodal hub correspond to discrete events (see Fig. 1) whereas traveller flows during the landing and the boarding phases present a continuous characteristic. Thus, to fully describe the functioning and the behaviour of public transport stations, they must be considered as a Hybrid Dynamic System (HDS)

Hybrid Petri nets, developed by David and Alla [1], have been used to model public stations (figure 2). This model was elaborated by considering that the boarding and the landing flows of travellers are independent and take place simultaneously.
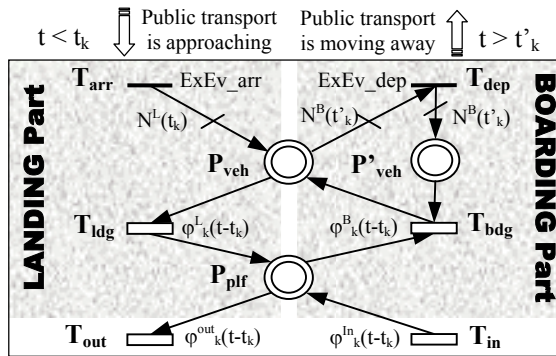

Figure 2:   Hybrid Petri net of public transport stations.

− **T$_{arr}$** (resp. **T$_{dep}$**) is a discrete transition synchronized on ExEv_arr (resp. ExEv_dep) which represents the discrete arrival (resp. departure) of the public transport at (resp. from) the station;

− **P$_{veh}$** is a continuous place which represents the vehicles of the public transport when it is well parked at the platform level;

− **P'$_{veh}$** is a continuous place corresponding to the maximal capacity (C$_{max}$) of the public transport;

−   **T_{ldg}** (resp. **T_{bdg}**) is a continuous transition which models travellers' landing (resp. travelers' boarding);

−   **P_{plf}** is a continuous place which represents the station platform (quay or stop);

−   **T_{out}** (resp. **T_{in}**) is a continuous transition which models the travellers' leaving the station toward the rest of the multimodal hub (resp. the travellers' arrivals from the rest of the multimodal hub into the station).

The different flows of this model are defined by the following equations and can be constant, linear, or exponential:

$$\forall k \in \{0, 1, 2, ..., K\} \text{ with } K \in \aleph,$$

$$\begin{cases} \forall t \in \left[t_k \quad t_k^{L-End} \right[, \ \varphi_k^L(t-t_k) \geq 0 \\ \forall t \in \left[t_k^{L-End} \quad t_{k+1} \right[, \ \varphi_k^L(t-t_k^{L-End}) = 0 \end{cases} \tag{1}$$

t_k^{L-End} indicates the end of the travellers' landing

$$\begin{cases} \forall t \in \left[t_k \quad t_k^{B-End} \right[, \ \varphi_k^B(t-t_k) \geq 0 \\ \forall t \in \left[t_k^{B-End} \quad t_{k+1} \right[, \ \varphi_k^B(t-t_k^{B-End}) = 0 \end{cases} \tag{2}$$

t_k^{B-End} indicates the end of the travellers' boarding

$$\forall t \in I_k = \left[t_k \quad t_{k+1} \right[, \quad \varphi_k^{out}(t-t_k) \geq 0 \text{ and } \varphi_k^{in}(t-t_k) \geq 0 \tag{3}$$

The amplitude of landing and boarding flows can be low or high according to the nature of the period (off-peak period or rush hours for instance).

The full analysis of this hybrid dynamic model is too long to be developed in this paper. We present a synthesis of this theoretical formalism in the following section.

4 Synthesis of the hybrid model analysis

The following results have been obtained by making the dynamic balances (differential equations) of the continuous places of the hybrid Petri net (Fig. 2) and then by integrating these differential equations.

In order to simplify this theoretical formalism, we have divided the whole analysis into two parts - the landing phase and the boarding phase - in accordance with the initial modelling assumptions.

4.1 Analysis of the travellers' landing phase

The landing phase begins at t_k when the public transport arrives at the station and may be completed for two main reasons: (i) all of the landing travellers into the public transport (N_k^L) have landed on the station platform at $t'_{vo,k}^L$, or (ii) the public transport leaves the station at t'_k. So, the variable t_k^{L-End} can be, according to the case, equal to $t_{vo,k}^L$ or t'_k.

a) **Evolution of the number of landing travellers (m^L_{veh}) into the public transport vehicles between two successive PT arrivals:**

$$\forall t \in \left[t_k \quad t_k^{L-End} \right[, \ m^L_{veh,k}(t-t_k) = N^L(t_k) - \int_{t_k}^{t} \varphi^L_k(u-t_k) \cdot du \tag{4-a}$$

$$\forall t \in \left[t_k^{L-End} \quad t_{k+1} \right[, \ m^L_{veh,k}(t-t_k^{L-End}) = 0 \tag{4-b}$$

b) **Evolution of the number of landed travellers (m^L_{plf}) on the station platform between two successive PT arrivals:**

$$\forall k \in \{0, 1, 2, ..., K\} \ \text{with} \ K \in \aleph,$$

$$\forall t \in \left[t_k \quad t_k^{L-End} \right[, \ m^L_{plf,k}(t-t_k) = \max \left[0, \left(m^L_{plf,k}(0) + F^L_{plf,k}(t-t_k) \right) \right] \tag{5-a}$$

$$\text{with} \begin{cases} m^L_{plf,k}(0) = \begin{cases} 0 \ \text{for} \ k = 0 \\ \lim\limits_{\substack{t \to t_k \\ t < t_k}} \left(m^L_{plf,k-1}(t-t_{k-1}^{L-End}) \right) \text{for} \ k > 0 \end{cases} \\ F^L_{plf,k}(t-t_k) = \int_{t_k}^{t} \left(\varphi^L_k(u-t_k) - \varphi^{out}_k(u-t_k) \right) \cdot du \end{cases}$$

$$\forall t \in \left[t_k^{L-End} \quad t_{k+1} \right[, m^L_{plf,k}(t-t_k^{L-End}) = \max \left[0, \left(m^L_{plf,k}(0) - F^{out}_k(t-t_k^{L-End}) \right) \right] \tag{5-b}$$

$$\text{with} \begin{cases} F^{out}_k(t-t_k^{L-End}) = \int_{t_k^{L-End}}^{t} \varphi^{out}_k(u-t_k^{L-End}) \cdot du \\ m^L_{plf,k}(0) = \lim\limits_{\substack{t \to t_k^{L-End} \\ t < t_k^{L-End}}} \left(m^L_{plf,k}(t-t_k) \right) \end{cases}$$

4.2 Analysis of the travellers' boarding phase

The boarding phase begins, as the landing phase, at t_k and may also stop for two reasons: (i) the maximum capacity of the public transport (C_{max}) is reached at $t^B_{vm,k} \le t'_k$, or (ii) the public transport leaves the station at t'_k.

So, the variable t^{B-End}_k can be, according to the case, equal to $t^B_{vm,k}$ or t'_k.

a) **Evolution of the number of boarded travellers (m^B_{veh}) into the public transport vehicles between two successive PT arrivals:**

$$\forall k \in \{0, 1, 2, ..., K\} \ \text{with} \ K \in \aleph, \forall t \in \left[t_k \quad t_k^{B-End} \right[\cup \left[t_k^{B-End} \quad t_{k+1} \right[,$$

$$m^B_{veh,k}(t-t_k) = \min \left[C_{max,k}, \left(\int_{t_k}^{t} \varphi^B_k(u-t_k) \cdot du \right) \right] \tag{6}$$

$$\text{if } \exists\, t = t_{po,k}^B \in \Big[t_k \quad t_k^{B-End}\Big[, \text{such th. } m_{plf,k}^B(t_{po,k}^B) = 0, \text{then } \forall t \in \Big[t_{po,k}^B \quad t_k^{B-End}\Big[,$$

$$m_{veh,k}^B(t - t_{po,k}^B) = \lim_{t \to t_{po,k}^B ,\, t < t_{po,k}^B} \Big(m_{veh,k}^B(t - t_k)\Big) \text{ while } \varphi_k^{in}(t - t_{po,k}^B) = 0$$

b) Evolution of the number of boarding travellers (m^B_{plf}) waiting on the station platform between two successive PT arrivals:

$$\forall k \in \{0, 1, 2, ..., K\} \text{ with } K \in \aleph,$$

$$\forall t \in \Big[t_k \quad t_k^{B-End}\Big[,\ m_{plf,k}^B(t - t_k) = \max\Big[0,\ \Big(m_{plf,k}^B(0) + F_{plf,k}^B(t - t_k)\Big)\Big] \quad (7\text{-}a)$$

$$\text{with } \begin{cases} m_{plf,k}^B(0) = \begin{cases} N_{wt,0}^B \ \text{ for } k = 0 \\ \lim_{\substack{t \to t_k \\ t < t_k}} \Big(m_{plf,k-1}^B(t - t_{k-1}^{B-End})\Big) \text{ for } k > 0 \end{cases} \\[2em] N_{wt,0}^B : \text{initial number of wainting travellers (platform)} \\[1em] F_{plf,k}^B(t - t_k) = \int_{t_k}^{t} \Big(\varphi_k^{in}(u - t_k) - \varphi_k^B(u - t_k)\Big) \cdot du \end{cases}$$

$$\forall t \in \Big[t_k^{B-End} \quad t_{k+1}\Big[,\ m_{plf,k}^B(t - t_k^{B-End}) = m_{plf,k}^B(0) + F_k^{in}(t - t_k^{B-End}) \quad (7\text{-}b)$$

$$\text{with } \begin{cases} m_{plf,k}^B(0) = \lim_{t \to t_k^{B-End} ,\, t < t_k^{B-End}} \Big(m_{plf,k}^B(t - t_k)\Big) \\[1.5em] F_k^{in}(t - t_k^{B-End}) = \int_{t_k^{B-End}}^{t} \varphi_k^{in}(u - t_k^{B-End}) \cdot du \end{cases}$$

We can observe that this formalism is complex because of the hybrid nature of public stations. These equations are piecewise continuous and most of them are non-linear. Thus, it is not easy to exploit this theoretical formalism in order to simulate some rush-hour period scenarios. That is why we suggest, in the following section, an algorithmic method which simplifies this formalism and makes it exploitable for design applications.

5 Analytical design algorithm

The analytical design algorithm presented in this section has two input arguments: the safety factor value ($fs \geq 1$) and the maximal travellers' density value (d_{max}). This algorithm synthesizes the whole theoretical analysis of the Hybrid Petri Net model. Moreover, it allows on e to follow the traveller numbers evolution into the public transport and on the station platform.

This algorithm has been written using the pseudo-code standard. The symbol % indicates the beginning and the end of comments.

AnalyticalDesignAlgorithm (*real* safety_factor**,** *real* max_trav_density**)**
BEGIN

% Data of the simulation%

% Time schedule and travellers' statistics%	% Initial flow values%
integer array N^L ← $[N^L_0, \cdots, N^L_k, \cdots, N^L_K]$ *integer array* N^W ← $[N^W_0, \cdots, N^W_k, \cdots, N^W_K]$ *real array* t ← $[t_0, \cdots, t_k, \cdots, t_K]$ *real array* t_prime ← $[t'_0, \cdots, t'_k, \cdots, t'_K]$	*real array* Φ^L ← $[\Phi^L_0, \cdots, \Phi^L_k, \cdots, \Phi^L_K]$ *real array* Φ^B ← $[\Phi^B_0, \cdots, \Phi^B_k, \cdots, \Phi^B_K]$ *real array* Φ^{in} ← $[\Phi^{in}_0, \cdots, \Phi^{in}_k, \cdots, \Phi^{in}_K]$ *real array* Φ^{out} ← $[\Phi^{out}_0, \cdots, \Phi^{out}_k, \cdots, \Phi^{out}_K]$
% Constant values% *integer* TIME_SCALE ← N *real* DELTA_T ← 1/TIME_SCALE *integer* CMAX ← value	% Initialisation of variables% *integer* i ← 0 *real array* land_veh_mark[0] ← 0 *real array* board_veh_mark[0] ← 0 *real array* land_plf_mark[0] ← 0 *real array* board_plf_mark[0] ← 0

for (0 ≤ *integer* k ≤ K)
  boolean lps ← TRUE% lps : landing phase state%
  boolean bps ← TRUE% bps : boarding phase state%
  integer index ← (t[k] – t[0]) * TIME_SCALE + 1
  real time_landing_end ← t_prime[k], *real* index_landing_end ← 0
  real time_boarding_end ← t_prime[k], *real* index_boarding_end ← 0
  for (t[k] ≤ *real* time ≤ t[k+1])% time increment is DELTA_T%
    % Flow integral computing: see equations [1] and [2]%
    real land_flow_integral ← LandFlowIntegralCompute("const", lps, k, t[k], time, Φ^L[k])
    real board_flow_integral ← BoardFlowIntegralCompute("const", bps, k, t[k], time, Φ^B[k])
    % Landing phase%
    if (lps = TRUE)
      % Flow integral computing: see equation [3]%
      real in_flow_integral ← InFlowIntegralCompute("const", k, t[k], time, Φ^{in}[k])
      real out_flow_integral ← OutFlowIntegralCompute("const", k, t[k], time, Φ^{out}[k])
      land_veh_mark[i] ← VehMarkComputeLpsTrue(land_flow_integral, N^L[k])
      land_plf_mark[i] ← PlfMarkComputeLpsTrue(k, land_plf_mark[index], land_flow_integral,↓
                                          ↑out_flow_integral)
      if ((land_veh_mark[i] ≤ 0) **OR** (time ≥ t_prime[k])) **then**
        lps ← FALSE
        index_landing_end ← i
        time_landing_end ← time
      endif
    else
      % Flow integral computing: see equation [3]%
      real in_flow_integral ← InFlowIntegralCompute("const", k, time_landing_end, time, Φ^{in}[k])
      real out_flow_integral ← OutFlowIntegralCompute("const", k, time_landing_end, time, Φ^{out}[k])
      land_veh_mark[i] ← VehMarkComputeLpsFalse
      land_plf_mark[i] ← PlfMarkComputeLpsFalse(out_flow_integral, ↓
                                   ↑ land_plf_mark[index_landing_end])
    endif
    % Boarding phase%
    if (bps = TRUE) **then**
      % Flow integral computing: see equation [3]%
      real in_flow_integral ← InFlowIntegralCompute("const", k, t[k], time, Φ^{in}[k])
      real out_flow_integral ← OutFlowIntegralCompute("const", k, t[k], time, Φ^{out}[k])
      board_veh_mark[i] ← VehMarkComputeBpsTrue(board_flow_integral, CMAX)
      board_plf_mark[i] ← PlfMarkComputeBpsTrue(k, t[0], t[k], TIME_SCALE, ↓
             ↑ board_plf_mark[index], board_flow_integral, in_flow_integral, N^W[k])
      if ((board_veh_mark[i] ≥ CMAX) **OR** (time ≥ t_prime[k])) **then**
        bps ← FALSE

```
            index_boarding_end ← i
            time_boarding_end ← time
        endif
    else
      % Flow integral computing: see equation [3]%
        real in_flow_integral ← InFlowIntegralCompute("const", k, time_boarding_end,time, Φⁱⁿ[k])
        real out_flow_integral←OutFlowIntegralCompute("const",k,time_boarding_end,time,Φᵒᵘᵗ[k])
        board_veh_mark[i] ← VehMarkComputeBpsFalse(time, t_prime[k], CMAX)
        board_plf_mark[i] ← PlfMarkComputeBpsFalse(in_flow_integral, ↓
                                          ↑ board_plf_mark[index_boarding_end])
    endif
    real array veh_mark[i] ← (land_veh_mark[i] + board_veh_mark[i])
    real array plf_mark[i] ← (land_plf_mark[i] + board_plf_mark[i])
    i ← i+1
  endfor
endfor
return real platform_surface ← safety_factor * max[plf_mark] / max_trav_density
```

END

% Details of the functions called in the algorithm%

VehMarkComputeLpsTrue(lfi , N^L) **BEGIN** *real* marking ← (N^L – lfi)% see eq. [4-a]% **return** marking **END**	**VehMarkComputeLpsFalse( )** **BEGIN** *real* marking ← 0% see eq. [4-b]% **return** marking **END**

PlfMarkComputeLpsTrue(k, limit, lfi, ofi)
BEGIN
if (k = 0) **then**
real const ← 0
else
const ← limit
endif
real lpm←(const + lfi –ofi)% see eq. [5-a]%
real marking ← max(0 , lpm)
return marking
END
	PlfMarkComputeLpsFalse(ofi, limit)
	BEGIN
	real const ← limit
	real lpm← (const –ofi)% see eq. [5-b]%
	real marking ← max(0 , lpm)
	return marking
	END

VehMarkComputeBpsTrue(bfi, C_{max})
BEGIN
real marking←min(C_{max} , bfi)% see eq. [6]%
return marking
END
	VehMarkComputeBpsFalse(t,t_prime,C_{max})
	BEGIN
	if (t < t_prime) **then**
	real marking = C_{max}% see eq. [6]%
	else
	marking = 0
	endif
	return marking
	END

PlfMarkComputeBpsTrue(k,limit,bfi,ifi, N^W)
BEGIN
if (k = 0) **then**
real const ← N^W
else
const ← limit
endif
real bpm← (const +ifi – bfi)% see eq. [7-a]%
real marking ← max(0 , bpm)
return marking
END
	PlfMarkComputeBpsFalse(ifi, limit)
	BEGIN
	real const ← limit
	real marking ← (const + ifi)% see eq.[7-b]%
	return marking
	END

6   Application: rush-hour period scenario

After a football match, most of the supporters take local trains or subways or trams, which connect the stadium station to the multimodal hub station in the town centre. Once arrived, supporters leave the hub station slowly because of the high concentration of travellers on the platform (queueing in front of the station outputs) and because most of them take time to comment on the match or to celebrate the victory. We will simulate this scenario with the following data:

Table 1:   Test data of the rush-hour period scenario.

CHARACTERISTICS OF THE LOCAL TRAINS			
PUBLIC TRANSPORT	**Train period**		**5** minutes
	Train composition		**20** vehicles with 3 inputs/outputs
	C_{max}		100 trav. per vehicle $\rightarrow$ **2000** travellers
TRAVELLERS' FLOWS	$\forall k,\ \Phi^L_k = \Phi^L$		1 trav. per 3 sec. per output $\rightarrow$ **1200** trav./min
	$\forall k,\ \Phi^B_k = \Phi^B$		**750** trav./min
	$\forall k,\ \Phi^{in}_k = \Phi^{in}$		**150** trav./min
	$\forall k,\ \Phi^{out}_k = \Phi^{out}$		**400** trav./min
TRAVELLERS' DISTRIBUTION PER TRAIN AT THE FINAL HUB STATION			
k	**t_k**	**t'_k**	**Number of landing travellers N^L_k**
0	0	3	2000
1	5	8	1900
2	10	13	1700
3	15	18	1500
4	20	23	1400

A rough design procedure would consist of considering that the maximal number of landing travellers ($N^L_{max}=C_{max}$) and the maximal number of travellers arriving at the platform: $\Phi^{in} \cdot (t_{k+1} - t_k)$ can be simultaneously on the platform. Then, the surface platform is given by (8), where f_s is the safety factor:

$$S_{plf} = f_s \cdot (N^L_{max}/d_{max}) = 1,2 \cdot (2750 / 5) = \mathbf{660}\ \mathbf{m^2} \qquad (8)$$

The Analytical design method estimates the maximal concentration of travellers on the platform through the maximal value of $m_{plf} = m^L_{plf} + m^B_{plf}$. Then, as previously, $m_{plf,max}$ is divided by d_{max} and multiplied by f_S, see (9).

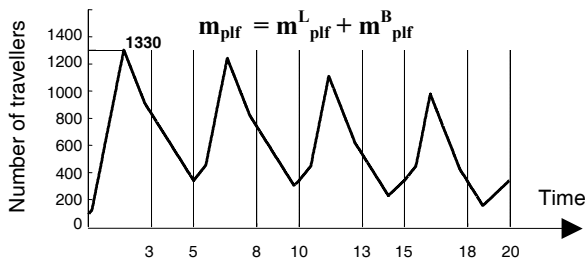$$S_{plf} = f_s \cdot (m_{plf,max} / d_{max}) = 1,2 \cdot (1330 / 5) \approx \mathbf{320}\ \mathbf{m^2} \qquad (9)$$

Figure 3:   Simulation of the rush-hour period scenario (Matlab).

We can observe a relative difference of 100 (340/660) ≈ 52% between the two calculated surfaces with this basic scenario. The space and cost saving is important and it would be more important by considering (i) a big multimodal hub, (ii) real data, and (iii) all kinds of public transport station (train, subway, tram, bus, etc.). The generalisation of this new design procedure can contribute to make public transport stations more secure with limited additional cost and space.

7   Conclusion

The analytical design formalism presented in this paper is based on a hybrid dynamic model of public transport stations which describes their functioning and their behaviour particularly when peaks of travellers are applied as input data. Owing to the *Analytical Design Algorithm* deduced from the model analysis, the travellers' concentration on station platforms can be assessed in order to simulate real or virtual scenarios and then design more precisely, and thus at a lower cost, secure platforms able to absorb peaks of travellers during rush-hour periods.

Acknowledgements

References

[1] David, R., Alla, H. On Hybrid Petri Nets. *Journal of Discrete Event Dynamic Systems: Theory and Applications*, **Vol. 11**, pp. 9-40, 2001.
[2] Durmisevic, S., Sariyildiz, E., A systematic quality assessment of underground spaces. *Cities*, **Vol. 18**, N° 1, pp. 13–23, 2001.
[3] Li, J.P., Train station passenger flow study. *Proceedings of the 2000 Winter Simulation Conference*, pp. 1173-1176, 2000.
[4] Luquet, F., Multimodality: urban hub management. *Revue RATP Savoir-Faire*, N° 26, pp. 31-35,1998.
[5] Moalla, M., Pulou, J., Sifakis, J., Réseaux de Petri synchronisés. *Revue RAIRO Automatique*, **Vol. 12**, n°2, pp. 103-130, 1978.
[6] Van Der Aalst, W.M.P., Analysis of Railway Stations by Means of Interval Timed Coloured Petri Nets. *Real-Time Systems*, **Vol 9**, pp 1-23. 1994
[7] Zaytoon, J., *Systèmes dynamiques hybrides*. Traité IC2. Hermès (Ed), Paris, 2001.
[8] ECS, http://www.cenorm.be/cenorm/index.htm