# Discrete-Time Control Analysis of Transport Channel Synchronization in 3G Radio Access Networks

Juan J. Alcaraz, Gaspar Pedreño, Fernando Cerdán, Joan García-Haro and Felipe García-Sánchez
Department of Information Technologies and Communications
Polytechnic University of Cartagena (UPCT)
Plaza del Hospital, 1, 30202 Cartagena, SPAIN
Telephone: +34 968336544. Fax: +34 968 32 53 38
{juan.alcaraz, gaspar.pedreno, fernando.cerdan, joang.haro@upct.es
felipe.garcia}@upct.es

**Keywords.** UTRAN, frame protocol, control system, synchronization, Iub

## Abstract
Transport channel synchronization is a function of 3G access networks that operates in the link between Radio Network Controllers (RNCs) and Base Stations (Nodes B) and is required to support macro-diversity in the downlink direction. Its objective is to assure that every frame sent by the RNC arrives at the Node B on time to be transmitted over the air interface avoiding excessive buffering at the Node B. This is achieved by means of a timing adjustment algorithm that tracks the delay of the link and corrects the sending time of the frames in the RNC. However, when the link experiences abrupt delay variations, e.g. because of a sudden traffic increment in an intermediate node, the classic algorithm may loss frames and, depending on its configuration and the transport delay, it can show an undesired oscillatory behaviour. In this paper we analyze the response of this mechanism under sudden delay increments, providing useful guidelines to configure it in order to prevent oscillations, avoiding excessive signalling and eventual frame losses. Moreover, we propose a simple algorithm that overcomes the limitations of the classic scheme. The configuration of our proposal is addressed by means of discrete-time control theory focusing on stability and performance considerations. We show the influence of the delay on these issues and present an adaptive strategy to make the system robust under delay variations.

## 1. Introduction
In UMTS Terrestrial Radio Access Networks (UTRAN), link-level communication between Nodes B and Radio Network Controllers (RNCs) through the Iub interface is regulated by the Frame Protocol (FP), specified by the 3GPP in [1] [2] and [3]. For dedicated channels (DCH), the Transport Channel Synchronization function of FP has the objective of delivering each frame to the Node B within a reception window which assures that the frame is transmitted over the air interface at its corresponding Transmission Time Interval (TTI) with the less possible buffering per channel at the Node B. Packet queuing should be done in the RNC, where the scheduling and the resource management processes are located.

The Iub interface is supported by an ATM or IP-based network, generally referred to as UTRAN transport network. The delay of the Iub depends on several factors, e.g. the distance of the links between the RNC and the Node B, the traffic intensity at each link, the amount of intermediate nodes, etc. This delay may experience abrupt variations in diverse situations:

- An intermediate node in the UTRAN transport network handling a large amount of traffic.
- When traffic engineering or path restoration mechanisms re-route Iub flows. See [6] and the references therein.
- If a source has been idle for a long time and the delay has experienced a noticeable drift that the synchronization process is not able to follow.

In the examples provided in [7], one-way delay of different Iub branches ranged from 12 to 31 ms considering voice traffic and ATM transport. The differences between Iub branches may be even higher for UTRAN-IP and data services, especially under heavy traffic.

Delay variations should be compensated by FP in order to assure that frames keep arriving within the reception window. This function is performed by the transport channel synchronization procedure and, more specifically, by the timing adjustment algorithm explained later in this paper. However, when delay variations are very steep, it takes some time to readjust the synchronization. During this time two facts affect the performance of the channels: (i) the signalling traffic in the Iub increases, (ii) frames arriving too far from the window boundaries are discarded, causing temporary performance degradation on the channels multiplexed over the Iub.

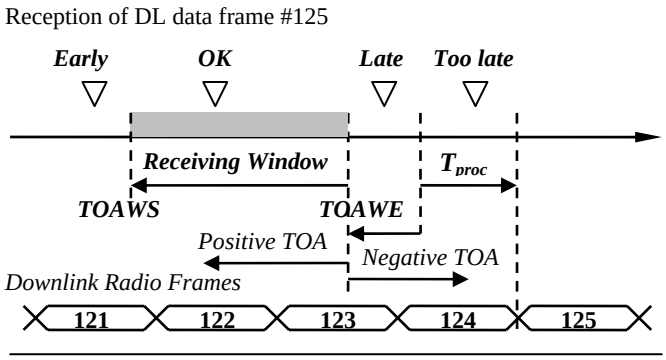
The duration of the degraded period after an abrupt delay variation depends on the timing adjustment algorithm implemented and on its configuration. Surprisingly, this issue has received little attention in previous research. The work in [5] is the only one, to the best of our knowledge, focused on transport channel synchronization. Authors propose a modification of the classical timing adjustment algorithm (CA). However its operation under abrupt delay increment is completely analogous to CA. Other works like [8], [9], [10] and [11] focus on the relationship between timing adjustment and Iub congestion. The starting point of these works is that delay variations provide information about the congestion in the Iub interface that may be used to flow-control data channels. However, all these papers assume that 3G nodes implement CA and no improvement to this scheme is suggested. The interest on transport channel synchronization is also present in [12] where it is discussed the impact of this issue on the design of QoS schemes for UTRAN-IP. All the aforementioned references come from laboratories and research groups of equipment manufacturers, which reveals the interest of the telecommunication industry in this issue.

In this paper we make the following contributions: First, we analyze the response of CA against steep delay increments. The results of this analysis are useful to configure the window size and the correction step size to prevent oscillations. Second, we propose a simple algorithm based on correcting the offset proportionally to the distance between the TOA and the centre of the window. This proposal can be analyzed by means of discrete-time control theory, which let us use classical techniques to design the gain ($K$) of the system assuring stability for different delay situations. In addition, we can determine $K$ from certain performance parameters, like the overshoot. In line with this control-based approach we refer to our proposal as Proportional Tracking Algorithm (PTA). Third, in order to assure stability under delay variations, we provide a simple procedure to reconfigure the gain according to the measured delay, resulting in an adaptive version of PTA. Fourth, theoretical results are validated through simulations. It should be noted that PTA is compatible with 3GPP specifications.

The rest of the paper is organized as follows. Section 2 analyzes the response of CA. Section 3 describes our proposal and its model as a discrete-time control system. Section 4 focuses on the stability and configuration of PTA, and presents its adaptive version. The validation of theoretical results through simulation is discussed in Section 5 including a performance evaluation under realistic traffic conditions in the Iub. Finally, Section 6 summarizes the contributions of this work.

## 2. Analysis of the classical algorithm

The transport channel synchronization procedure operates on each transport channel associated to a DCH in the downlink direction and relies on an estimation of the downlink delay (*offset*). If the *offset* does not differ too much from the actual delay, each frame is expected to arrive at the Node B within a reception window. If a frame arrives outside this window, the Node B responds with a Timing Adjustment (TA) frame containing the Time Of Arrival (TOA) of the received frame. The RNC uses this feedback information to adjust the *offset* and thus, the sending time instant of the next frame, trying to steer the arrival time of the frames into the window. This procedure is known as timing adjustment. If a frame arrives so late that it cannot be processed before its TTI, the frame is discarded. Fig. 1 shows the possible frame reception instants and the names of the temporal references. For a more detailed explanation of the related signalling the reader is referred to [4].




| | |
|---|---|
| TOA | Time Of Arrival |
| TOAWS | TOA Window Startpoint |
| TOAWE | TOA Window Endpoint |
| $T_{proc}$ | Processing time before transmission over air interface |

Fig. 1. Reception window in the Node B.

The classical timing adjustment algorithm (CA) basically adds or subtracts a constant value, *K*, to the *offset* if the TOA indicates a late or an early arrival respectively. It is frequent to see this algorithm referred to the sending time instant [5]:

$$\begin{aligned} t_{n+1} &= t_n + t_{TTI} - K \quad if\ TOA < 0 \\ t_{n+1} &= t_n + t_{TTI} + K \quad if\ TOA > TOAWS \end{aligned} \tag{1}$$

where $t_{TTI}$ is the duration of a TTI and $t_n$ is the transmission time instant of frame *n*. This algorithm can also be expressed in terms of offset corrections with the following equation:

$$\begin{aligned} offset_{n+1} &= offset_n + K \quad if\ TOA < 0 \\ offset_{n+1} &= offset_n - K \quad if\ TOA > TOAWS \end{aligned} \tag{2}$$

The main advantage of CA is its simplicity. However, some aspects of this algorithm have not been fully investigated, e.g. how to suitably configure the step *K* and the window size considering the inherent oscillatory behaviour of the algorithm.

Because the signals involved in timing adjustment (the delay, the error and the offset), take one value per frame, we can model the CA algorithm as a discrete-time control system making the following assumptions. First, the source transmits one frame per TTI, i.e. there are no idle times for data traffic in the period under study. Second, the inter-departure time between consecutive frames in the RNC is constant and equal to $t_{TTI}$. Therefore we neglect the variation in inter-departure time caused by the offset correction done when TA frame are received. These assumptions are validated by means the simulator presented later in Section 6.
Fig. 2 illustrates the operation of the time adjustment mechanism in a late arrival situation. This figure shows the delays of the system and the correction (Δ) applied to the offset, which equals to the time advance applied to the transmission time instant of frame *n*. According to (2), $\Delta = K$ if $TOA < 0$ and $\Delta = -K$ if $TOA > TOAWS$. We define the Round Trip Time (RTT) of the system as the time elapsed between the sending time of a frame and the moment when the TA frame is applied to another frame. The number of TTIs in the RTT is $r = \lceil RTT / t_{TTI} \rceil$. Defining *j* and *m* as the downlink and uplink delay in TTIs respectively, $r = j + m$.

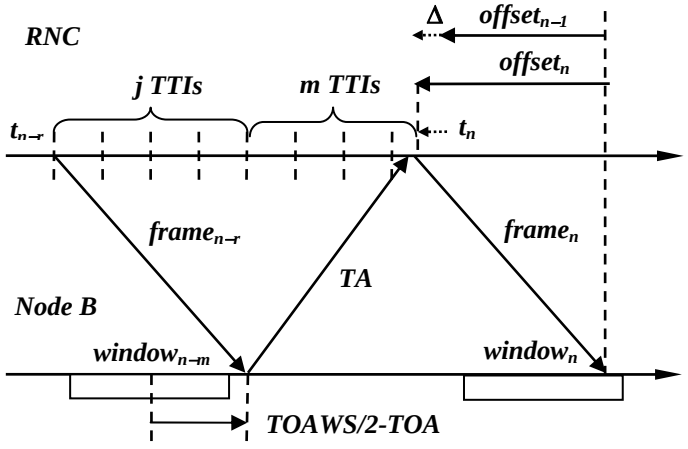



Fig. 2.Temporal diagram of the offset adjustment algorithm.

If the delay predicted for a frame equals the delay experienced for this frame when it arrives at the node B, we can assume that this frame will arrive at the centre of the reception window, considering that the correction algorithm tends to move the arrival time instant towards the centre of the window whenever a frame arrives outside the window. Therefore, as shown in Fig. 2, if frame $n-r$ arrives outside the window, this means that the offset associated to frame $n-r$ ($offset_{n-r}$) was smaller than the actual delay experienced by that frame, measured in time-slot $n-m$ ($delay_{n-m}$), in a magnitude greater than *TOAWS/2*. As shown in the figure, the error of the offset respect the delay is given by *TOAWS/2* – *TOA*, which is equivalent to $e_n = delay_{n-m} - offset_{n-r}$. Let us

define a convenience function $f(e_n)$, such that $f(e_n) = 1$ if $e_n > TOAWS/2$ (early arrival), $f(e_n) = -1$ if $e_n < -TOAWS/2$ (late arrival), and $f(e_n) = 0$ (in-window arrival), otherwise. With this terminology we can express (2) as

$$offset_{n+1} = offset_n + K \cdot f(e_n) \quad (3)$$

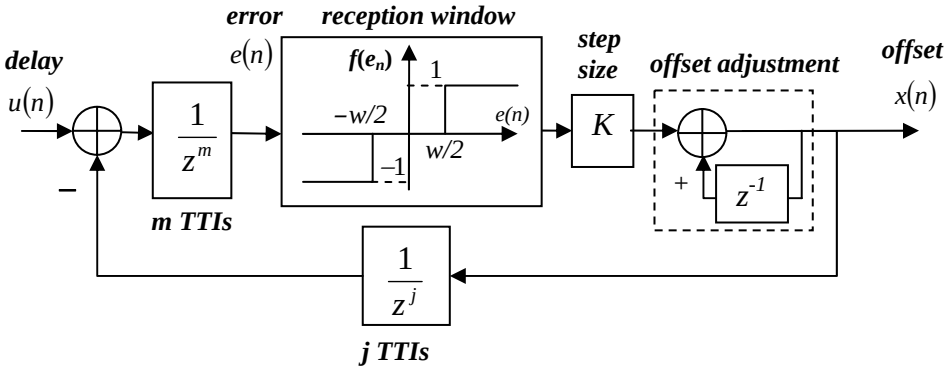



Fig. 3. Discrete-time control model of the classic algorithm.

Fig. 3 shows a diagram of the control system described by the difference equation (3). Note that the transfer function of the dashed box is $z/(z-1)$. Blocks $1/z^j$ and $1/z^m$ represent respectively the downlink and uplink delay. The signal $u(n)$, corresponds to $delay_n$ and acts as the control signal, $x(n)$ is $offset_n$, and $e(n)$ is $e_n$. The effect of the window is modelled with a block whose response function is $f(e_n)$, i.e. its output is 0 if $|e_n| < w/2$, where $w$ equals $TOAWS$, and 1 otherwise. Because of the nonlinear nature of $f(e_n)$ we can not apply classical control analysis for designing the step value $K$, that we refer to as the gain of the system CA. Considering the particular case $w = 0$, $f(e_n) = sign(e_n)$. Applying this notation, equation (3) is given by

$$x(n) = x(n-1) + K\,sign(u(n-m) - x(n-r)). \quad (4)$$

The simplification $w = 0$ represents a situation where the RNC receives information of the arrival time of every frame, since a TA frame is sent for every arrival. As shown later in the analysis, this simplification is useful to: a) provide insight in the inherent oscillatory nature of CA, b) to analyze the stability of the algorithm PTA, proposed in section 4 and c) to compare the performance of CA and PTA theoretically.

Although CA has been referenced several times in previous works, no one has given a methodology for its configuration so far. The following analysis is intended to contribute on this. In agreement with the scenario described in Section 1 and in the analysis of PTA developed later in Section 5, we study CA under a step signal as control input $u(n)$, i.e. when the Iub experiences an abrupt delay increment. Without loss of generality, let us assume that $u(n) = 0$ for $n < 0$ and $u(n) = C$ for $n \geq 0$. We consider two cases for the difference equation describing CA, $w = 0$ and $w \neq 0$.

a) Case 1. $w = 0$.

*Proposition 1*

Given the algorithm described by the difference equation (4), with the following initial conditions:

$x(n) = 0$ for $n \leq 0$, $u(n) = C$ for $n \geq 0$ and $u(n) = 0$ for $n < 0$, where we consider two cases for $C$:

Case 1.1. $C = aK$, where $a$ is a positive integer.
Case 1.2. $C = aK + b$, where $b$ is a positive real number such that $b < K$.
The sequence $x(n)$ grows linearly until reaching the value $aK$, and from this moment it oscillates with a period length of $4r - 2$ time-slots. The maximum and minimum amplitude values are:
Case 1.1. The maximum is $aK + (r-1)K$ and the minimum $aK - (r-1)K$.
Case 1.2. The maximum is $aK + rK$ and the minimum $aK - (r-1)K$.

b) Case 2. $w \neq 0$.
The difference equation for CA with $w \neq 0$ is the following:

$$x(n) = x(n-1) + K\left(U\left(x(n-r) - u(n-m) - w/2\right) - U\left(u(n-m) - x(n-r) - w/2\right)\right) \quad (5)$$

Where $U(\cdot)$ is the unit step function (not to be confused with the input signal $u(n)$).

*Proposition 2*
Given the algorithm described by the difference equation (5), with the following initial conditions:
$x(n) = 0$ for $n \leq 0$, $u(n) = C$ for $n \geq 0$ and $u(n) = 0$ for $n < 0$, $K < C$, $w = qK + \alpha$, with $q$ a positive integer and $\alpha$ a positive real number such that $\alpha < K$. There are two different responses of the sequence $x(n)$, determined by two different cases for $C$:
Case 2.1. $C - w/2 = aK$, where $a$ is a positive integer. The sequence $x(n)$ grows linearly until reaching the value $aK + (r-1)K$, which is the steady-state value if $w > (r-1)K$. Otherwise, if $w \leq (r-1)K$, the sequence $x(n)$ will oscillate with a period length of $4r - 2$ time-slots, maximum amplitude equal to $aK + (r-1)K$ and minimum amplitude equal to $aK - (r - q - 1)K$.
Case 2.2. $C - w/2 = aK + b$, where $b$ is a positive real number such that $b < K$. The sequence $x(n)$ grows linearly until reaching the value $aK + rK$, which is the steady-state value if $w > rK - b$. Otherwise, if $w \leq rK - b$, the sequence $x(n)$ will oscillate with a period length of $4r - 2$ time-slots, maximum amplitude equal to $aK + rK$ and minimum amplitude equal to $aK - (r - q - 1)K$.

*Proposition 3.* In both algorithms, described by equations (4) and (5), with initial conditions $u(n) = C$ for $n \geq 0$, and $u(n) = 0$ for $n < 0$ and $x(n) = 0$ for $n < 0$, the sequence $x(n)$ reaches the value $hK$ for the first time in $h + m$ time-slots, with $h$ a positive integer, such that $hK \leq A$, where $A$ is the maximum achievable amplitude of $x(n)$.

The proofs of the three propositions above are given in Appendix A. Proposition 3 can be used to design $K$ for a given speed of the step response of the system. Proposition 2 gives a useful condition to configure the window size and the value of $K$ in order to prevent the response to oscillate, considering the delay of the system. This is an interesting result because it shows that the window does not only avoid excessive signalling in FP, but also alleviates the inherent oscillatory nature of CA, shown in Proposition 1. Proposition 2 also shows that, in absence of oscillations, the window causes an error in $x(n)$ with respect to $u(n)$ equal to $(r-1)K - w/2$. With these considerations, we will use Case 1 in order to compare CA and PTA in Section 5.
It should be noted that oscillations in the algorithm degrade its performance because of two reasons. First, the signalling load in the Iub interface increases unnecessarily. Second, if the amplitude of the oscillation is too large, it may cause frame losses because of late frame arrivals or buffer overflow in the Node B if many frames arrive too soon.

## 3. Description of the proposal

Our proposal is based on a classical control model. As explained in Section 1, the transmission time instant of each frame is determined by the offset at the RNC side. The difference between the offset and the actual downlink delay equals, as previously discussed, *TOAWS/2 – TOA*, which we refer to as the error. The proposed strategy is to correct the offset proportionally to this error, in other words, to use a proportional tracking system. The sending time instant is computed according to the following expression, upon the reception of a TA frame:

$$t_{n+1} = t_n + t_{TTI} + K(TOA - TOAWS/2)\,. \quad (6)$$

PTA is fully described by the equation above. For example, if a frame arrives "late", the factor *TOA – TOAWS/2* in (6) is negative; therefore next frame will be scheduled in advance, implying an equivalent increment in offset estimation.

### *4.1. Model based on discrete-time control theory*

In order to analyze the PTA algorithm described in (6), we use the simplification $w = 0$, accordingly to Section 3, obtaining a completely linear system that can therefore be analyzed by means of classical discrete-control theory. This assumption represents a worst case scenario in terms of stability, which is our main concern in the analysis. This is also in accordance with the results of CA, where the window was shown to have a stabilizing effect compared to the $w = 0$ scenario. According to the definitions given in section 3, and considering the offset correction, we can express (6) as

$$offset_{n+1} = offset_n + K\,e_n\,, \quad (7)$$

The time diagram in Fig. 2 is also valid for the algorithm described by (7), considering that the correction applied to the offset in PTA is $\Delta = K \cdot e_n$. Fig. 4 shows a diagram of the control system for PTA. Note that this model is basically the model in Fig. 3 without the nonlinear block $f(e_n)$.

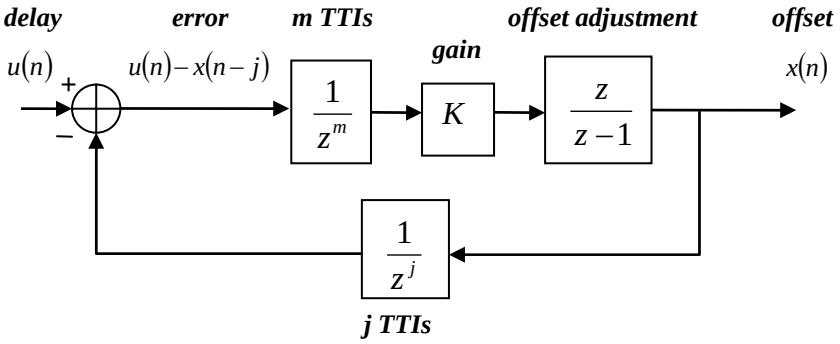



Fig. 4. Discrete-time control model of PTA.

The transfer function of the forward path of the model is:

$$G(z) = \frac{K \cdot z}{z^m (z-1)}\,. \quad (8)$$

And for the feedback loop we have:

$$H(z) = \frac{1}{z^j}. \quad (9)$$

Considering $U(z)$ and $X(z)$ as the z-transforms of $u(n)$ and $x(n)$ respectively, the open-loop transfer function is $G(z)H(z)$, and the closed-loop transfer function is:

$$\frac{X(z)}{U(z)} = \frac{G(z)}{1+G(z)H(z)}. \quad (10)$$

We can check this expression by writing (7) as a difference equation:

$$x(n) = x(n-1) + K\big(u(n-m) - x(n-r)\big). \quad (11)$$

Taking z-transforms in (11) we get

$$X(z) = z^{-1}X(z) + K\cdot z^{-m}U(z) - K\cdot z^{-r}X(z), \quad (12)$$

That gives the following transfer function:

$$\frac{X(z)}{U(z)} = \frac{K\cdot z^j}{z^r - z^{r-1} + K}, \quad (13)$$

which can be also obtained by substituting (8) and (9) in (10).
This model is very useful to study the system stability and to design the gain $K$ from given quality parameters.

## 4. Analysis and Configuration of PTA

In this section we describe the configuration of the gain ($K$) in PTA. Firstly, we find the critical $K$ value, $K_c$, above which the system is unstable. Performing the Jury stability test [13] for different values of $r$ we obtain the following critical gain values.

Table I. $K_c$ for different values of $r$.

| $r$ | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|
| $K_c$ | 1 | 0.618 | 0.445 | 0.347 | 0.285 | 0.241 |

It is clear that, since the degree of the transfer function of PTA is determined by the delay $r$, a longer delay imposes a smaller $K_c$. Therefore, the gain should be configured below the critical gain for the maximum expected delay of the system in order to avoid instability of the algorithm. However, a better strategy is to automatically adjust the gain according to the estimation of the downlink delay so that the system never enters in the unstable region. This strategy is developed in the next subsection.

### *4.1. Computation of the gain (K) from performance objectives*

In classical control design, the gain $K$ is usually determined form performance parameters that specify the performance of the step response in the time domain. Because, as explained in Section 4, oscillations may cause frame losses, we choose the overshoot as the main performance parameter. The overshoot ($M_p$) is defined as the ratio between the maximum amplitude the response (amplitude of its first peak) and the amplitude of the input step signal. In our case, the overshoot corresponds to the overestimation of the offset. Therefore, a large overshoot implies that the system may lose frames during the transient period. Considering a step input of the form $u(n) = C$ for $n > 0$, and an output which first peak equals ($A_{max}$), the definition is:

$$M_p(\%) = \frac{A_{\max}}{C}\cdot 100. \quad (16)$$

According to classical control theory [13], the above equation is related to the damping ratio parameter ($\zeta$) by

$$\frac{A_{\max}}{C} = e^{-\zeta\pi/\sqrt{1-\zeta^2}} \text{ . (17)}$$

Using (17) we obtain the ratio $\zeta$ for a given overshoot. The gain $K$ is obtained by solving the characteristic equation $1+G(z)H(z)=0$, which can be expressed as $F(z)=-1$, where $F(z)=G(z)H(z)$. Given (8) and (9), we have:

$$F(z) = \frac{Kz}{z^r(z-1)} = -1 \text{ . (18)}$$

According to [13], if $\zeta < 1$, the poles of the system are related to the damping ratio ($\zeta$) with the following expression $z = \exp\left(-2\pi\zeta\omega/\sqrt{1-\zeta^2} + j2\pi\omega\right)$, where $\omega = \omega_d/\omega_s$, $\omega_d$ is the transient oscillation frequency and $\omega_s$ is the sampling frequency in rad/s. In our case $\omega_s = 2\pi/t_{TTI}$. Given that $F(z)$ is a complex magnitude, (18) can be split into two equations, one for the phase of $F(z)$, equal to $\pi$ radians, and another for the module, which must equal 1. The phase condition yields the equation:

$$(r-1)2\pi\omega + \arg(C^{\omega}e^{j2\pi\omega} - 1) = \pi \text{ , (19)}$$

where $C = e^{-2\pi\zeta/\sqrt{1-\zeta^2}}$. From (19) we obtain $\omega$, which is applied to the module condition to obtain $K$:

$$K = C^{\omega(r-1)}\left|C^{\omega}e^{j2\pi\omega} - 1\right| \text{ . (20)}$$

Table II shows $K$ for different overshoots and delay values.

Table II. Gain ($K$) for different overshoot percentages.

| $r$ | 2% ($\zeta$=0.78) | 5% ($\zeta$=0.69) | 10% ($\zeta$=0.59) | 15% ($\zeta$=0.52) | 20% ($\zeta$=0.45) |
|---|---|---|---|---|---|
| 2 | 0.3133 | 0.3464 | 0.3810 | 0.4285 | 0.4649 |
| 3 | 0.1868 | 0.2071 | 0.2340 | 0.2578 | 0.2803 |
| 4 | 0.1332 | 0.1478 | 0.1671 | 0.1843 | 0.2005 |
| 5 | 0.1035 | 0.1149 | 0.1210 | 0.1434 | 0.1560 |
| 6 | 0.0847 | 0.0940 | 0.1063 | 0.1173 | 0.1277 |
| 7 | 0.0716 | 0.0795 | 0.0810 | 0.0993 | 0.1081 |

As expected, for a fixed $\zeta$, the gain decreases for smaller $r$ values. This effect is completely analogous to the reduction of the critical gain when the delay increases. The adaptive strategy consists of automatically adjusting the gain $K$ to the estimated $r$ value. Since $r$ is the RTT of the Iub link in time-slots, when $r$ changes, the offset measured by the tracking algorithm at the RNC changes as well, since the offset is in fact an estimation of the downlink delay. Therefore, the RTT of the Iub link can be approximated as 2×*offset*, which is a worst-case scenario in most situations given that the downlink delay is, in general, larger than the uplink delay [7]. According to this RTT estimation, the RNC adjusts $K$. This adjustment can be based on a list of pre-calculated values for a desired performance, e.g. the column for 10 % overshoot in Table II. Another feasible option is to use a function, *K(RTT)*. One way to obtain this function is to use a least square fit to approximate a set of gain values. The following function was found to be able to approximate the values of $K$ for a given overshoot:

$$K(t) = a + b\cdot t + c\cdot e^{d\cdot t} \text{ (21)}$$

Where $t$ is the RTT estimation and $a$, $b$, $c$, and $d$ are the coefficients that must be obtained by means of a least squares fit from the set of gain values that provide the desired overshoot. Considering an overshoot of 10%, the least square solution is given by $a = 0.1074$, $b = -0.4047$, $c = 1.1201$ and $d = -67.8995$. These coefficients remain fixed during the operation of the algorithm, assuring a similar response of the system under any delay conditions.

It may be argued that, when Iub delay variations occur, the estimated RTT ($2\times$*offset*) differs temporally from the real RTT value because the algorithm tracks the real value with a certain delay. This is not a problem provided that the selected overshoot is sufficiently conservative. The simulations done for this paper have shown that a 10% overshoot is a safe value in terms of frame loses and stability allowing the algorithm to present similar performance independently of the delay. Besides, this auto-adaptive gain only implies one algebraic operation per each offset correction.

*4.2 Comparison of the theoretical step responses*

We use the step response of the algorithms in order to compare performance. For instance, setting the overshoot percentage $M_p = 10\%$, we determine the gain $K$ for PTA as explained above. In CA, we consider the overshoot as the maximum oscillation amplitude ($A_{max}$), therefore

$$100 \cdot (A_{\max} - C)/C = 100 + M_p \text{ . (22)}$$

Following with CA, we assume that $K$ adapts to the delay $r$, i.e. $K$ is computed for each $r$ value according to Proposition 1. Note that $r = 2$ and $r = 4$ correspond respectively to Case 1.1 and 1.2. Because in CA $K$ is a magnitude to be added or subtracted to the offset it should be expressed in time units (*ms*). Let us define the rising time ($t_r$) as the time to reach the closest value to $C$ for the first time. For the CA, this value is given by Proposition 3:

$$t_r = \left(\lfloor C/K \rfloor + m\right) t_{TTI} \text{ . (23)}$$

Similarly, for the PTA the rising time is defined as the time to achieve the final or steady-state value for the first time and, according to [13], is given by:

$$t_r = \frac{\pi - \tan^{-1}\left(\dfrac{\omega_d}{\zeta\omega_n}\right)}{\omega_d}, \text{ (19)}$$

where $\omega_n$ is the natural oscillation frequency and is given by $\omega_n = \omega_d / \sqrt{1-\zeta^2}$ . The rising time gives an idea of the responsiveness of the algorithm, i.e. how fast the algorithm can recover the synchronism after a delay change in the Iub. Table III compares the rising time of CA and PTA for $M_p = 10\%$, considering two RTT values, $r = 2$ and $r = 4$. The input amplitude is $C = 10$ ms.

Table III. Rising time for CA and PTA.

| *r* | *Parameters* | *CA* | *PTA* |
|---|---|---|---|
| *2* | *K* | *1(ms)* | *0.390* |
| | *Rising time, $t_r$* | *110 ms* | *34.3 ms* |
| *4* | *K* | *0.323 (ms)* | *0.167* |
| | *Rising time, $t_r$* | *320 ms* | *81.80 ms* |

Fig. 5 compares the responses of both algorithms to input signals of 10 ms and 20 ms and RTTs given by $r = 2$ and $r = 4$. The concepts of overshoot and rise time are clearly illustrated in this figure. See that an additional benefit of PTA is that quality parameters remain constant independently of the input step magnitude.

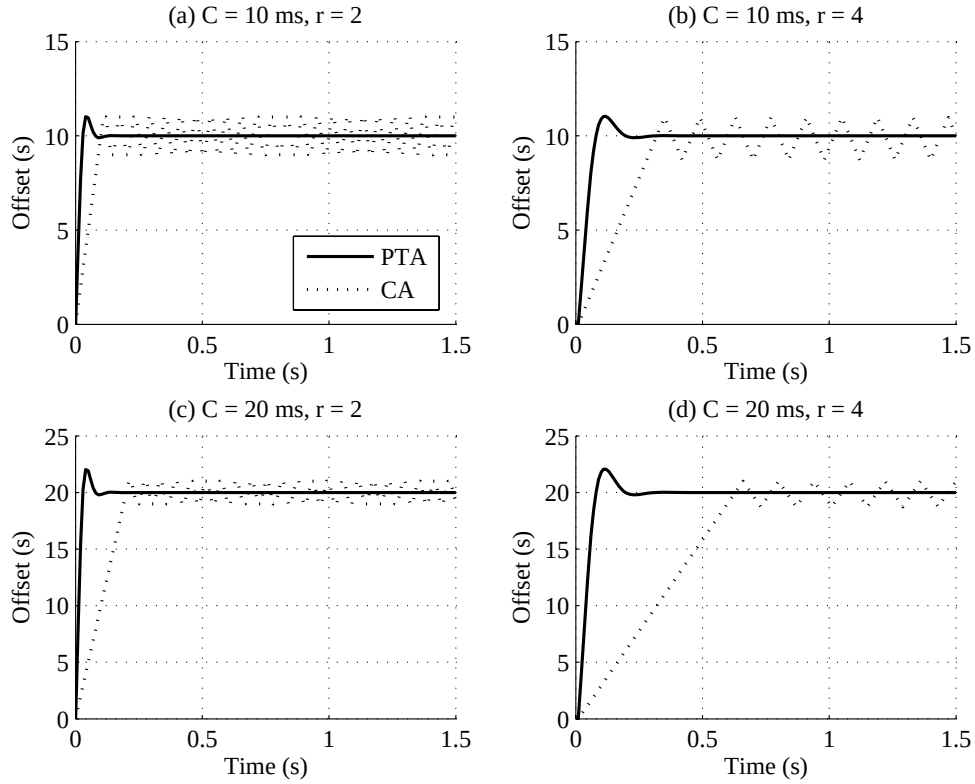



Fig. 5. Step responses of CA and PTA.

## 5. Simulation Results

In this section we compare the performance of the classic timing adjustment algorithm with our proposal in terms of transport network performance by means of an UTRAN transport network simulator, which was also used to validate the mathematical results described in previous sections.

### *5.1. Simulator description*

The simulator used for this paper has been fully developed in OMNeT++ [14], and comprises a detailed implementation of the Frame Protocol, the required Node B and RNC functions, an Iub link and a set of data and speech sources. Details about the sources are given in subsection 6.3. Fig. 6 shows the structure of the simulated system.

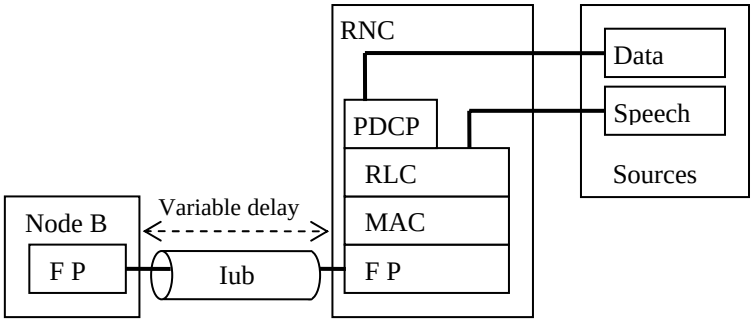



Fig. 6. Simulator structure.

The bandwidth of the Iub interface was set to 1,7 Mbit/s, roughly equal to the net rate of an E1 PDH link discounting 10% for traffic signalling and O&M. The Iub delay is variable in order to study the reaction of the algorithms to sudden delay increments within controlled conditions.

*5.2. Comparison with the classical scheme*

It is especially interesting to observe how PTA performs compared to the classic algorithm in terms of packet losses and signalling overhead, which are the performance parameters that our proposal aims to improve. For this comparison we have considered an Iub delay change from 10 ms to 50 ms, after which we have registered the instants when TA frames are sent ("signalling events") and the instants when frames arrive too late ("loss events"). The CA scheme is configured to $K$ = 1 ms and $K$ = 3 ms, and PTA is configured for a 10 % overhead. The window size is set to 10 ms in every scheme. Fig. 7 shows the traces of this simulation, including the evolution of the TOA to illustrate the relation between the response of the system and the registered events. As expected, since PTA reduces the synchronism recovery time, it also experiences less signalling and loss events than CA. Compared to CA configured to $K$ = 1 ms, PTA reduces frame losses after the delay increase by more than 60 %. If the gain of CA is set to $K$ = 3 ms, the recovery time of CA is smaller but the system presents an oscillatory behaviour causing periodic frame losses. Note that our theoretical model for CA predicted and explained this behaviour.

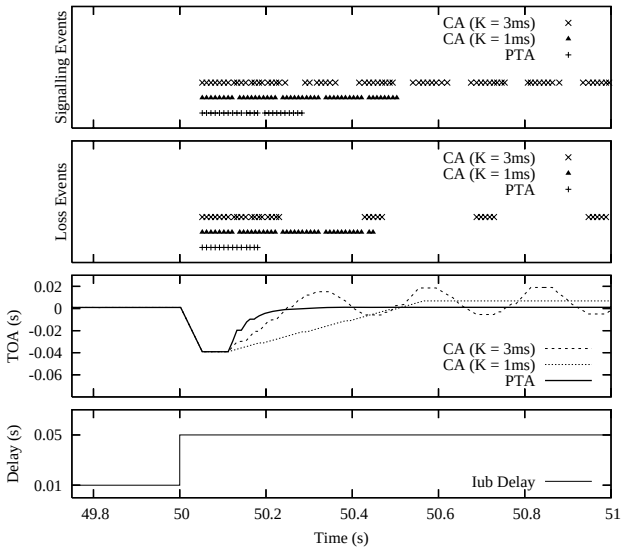



Fig. 7. Reaction of both algorithms to a delay increment.

*5.3. Performance under realistic traffic situation*

In this subsection, both algorithms are evaluated under realistic traffic conditions. In the simulated scenario, the Iub interface contains all the traffic flows handled by a typical Node B. (30 speech flows and 6 data flows). Speech sources are based on Adaptive Multi-Rate (AMR) codecs in the 12.2 kbps mode. Each data source uses a 384 Kbit/s DCH and is associated to a WWW application, consisting of a sequence of file downloads. The IP packet size is set to 1500 bytes. For a more detailed description of the used data and speech sources the reader is referred to [15].

We are mainly interested in the ratio between the number of frames arriving "too late" and the total number of frames received at the Node B (*loss ratio*). It is also interesting to assess the ratio between the number of TA frames sent by the Node B and the total number of data frames received at the Node B (*signalling ratio*). The simulation tests consists of abrupt variations of the delay experienced by an Iub interface. The nominal Iub delay is 50 ms, and the delay variations range form –40 ms (delay decrement) to 40 ms (delay increment).The performance is evaluated during a period of 60 seconds after the delay variation. Each performance figure is obtained by averaging 10 simulation runs and the confidence intervals are obtained with confidence degree of 95% according to a t-student distribution. TOAWS and TOAWE are set to 10 ms and 5 ms respectively.

The loss ratio and the signalling ratio for Iub delay variations are shown in Fig. 8 and Fig. 9 respectively. These performance figures are in accordance with the results observed in previous subsection, where only a single FP flow was considered. These results illustrate the practical performance implications of the timing adjustment algorithm and its configuration.

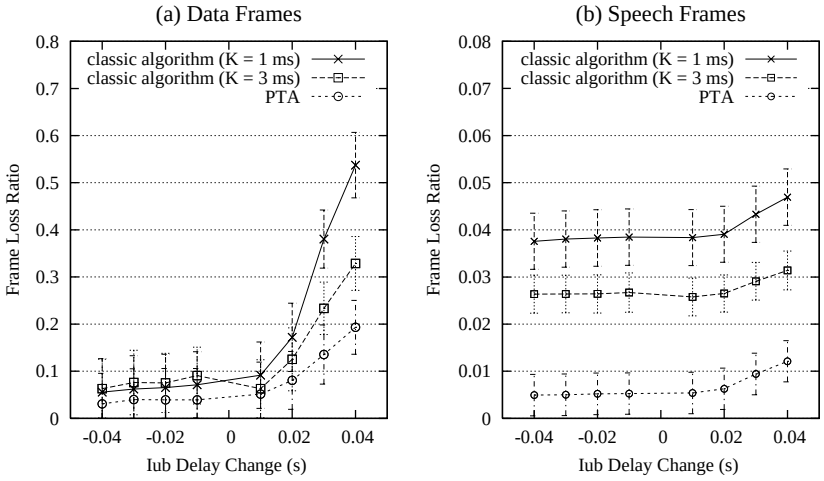



Fig. 8. Loss Ratio under realistic traffic conditions.

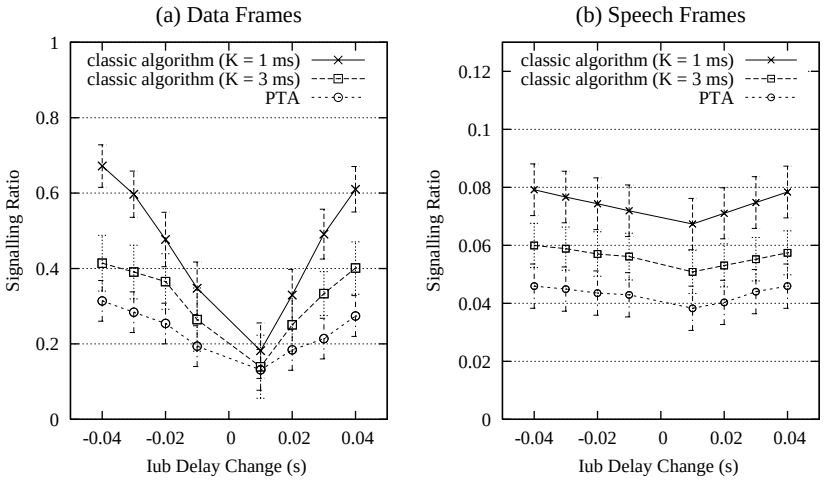



Fig. 9. Signaling Ratio under realistic traffic conditions.

## 6. Conclusions

In this paper, the transport channel synchronization mechanism in UTRAN has been analyzed for the first time. We have shown the inherent oscillatory nature of the classic timing adjustment algorithm, which can only reach a steady-state value thanks to the reception window. However, oscillation may appear even with a window size greater than zero. We have disclosed the relation among the window size, the Iub RTT and step size *K* that determines the conditions that make the system oscillate, as well as the cycle amplitude and period.

On the other hand, we have proposed a new mechanism (PTA) based on control theory that corrects the offset proportionally to the time distance between the instant of a frame arrival and the centre of the window. Modelling PTA as a discrete-time control system we have used a classical control approach to configure the gain *K* to assure stability and to comply with quality parameters of the step response like the overshoot. Because the RTT has a great impact on stability, the gain should be adjusted accordingly to delay variations. We provide a closed expression relating *K* with the RTT, resulting in an adaptive version of PTA. Simulation experiments have shown that the theoretical results give an accurate prediction of worst-case performance figures and finally the performance was assessed under realistic traffic conditions in the Iub interface, confirming that the control-based strategy is an effective technique to improve performance in wireless cellular transport networks.

## Annex A. Mathematical Proofs

In order to simplify the notation, $x(n) \equiv x_n$ and $u(n) \equiv u_n$.

*1. Proof of Proposition 1*

Classic Algorithm is given by the difference equation $x_n = x_{n-1} + K\,sign(u_{n-m} - x_{n-r})$. Considering an input signal $u_n = C$ for $n \geq 0$, from the time-slot when $u_{n-m} = C$, we can express CA as $x_n = x_{n-1} + K\,sign(C - x_{n-r})$. From this equation, we can directly derive the following useful properties of the output sequence:

(i) $x_{n+m} = x_n + Km \quad for\ 0 \leq m < r, \quad if\ x_{n-i} < C \quad for\ 0 < i < r$

(ii) $x_{n+m} = x_n - Km \quad for\ 0 \leq m < r, \quad if\ x_{n-i} > C \quad for\ 0 < i < r$

(iii) $x_n = x_{n-1} \quad if \; x_{n-r} = C$

*1.1. Proof of Case 1.1, $C = aK$*

Assuming that at time-slot $n - r$ we have $x_{n-r} = C - rK$, such that $x_{n-r}$ satisfies conditions of property (i), the output sequence increases by *K* each time-slot. Property (i) hols for $x_n = aK$ but not for $x_{n+1}$, therefore the sequence increases until reaching its maximum value $x_{n+r-1} = aK + K(r-1)$. Applying (iii) $x_{n+r} = x_{n+r-1}$. See that $x_{n+r}$ satisfies (ii), therefore the sequence starts to diminish by *K* each time-slot. In $x_{n+2r-1} = aK + K(r-1) - K(r-1) = aK$ the property (ii) applies, but not in $x_{n+2r}$, therefore the sequence diminishes until a minimum value at $x_{n+3r-2} = aK - K(r-1)$. Applying (iii), $x_{n+3r-1} = x_{n+3r-2}$. Because $x_{n+3r-1}$ satisfies conditions in (i), and its value equals $x_{n-r+1}$, where (i) also applies, the sequence of output values restarts from this point, and does so periodically. By subtracting time-slot indexes we the length of the cycle $p = n + 3r - 1 - (n - r + 1) = 4r - 2$. The maximum output value within the cycle is $aK + K(r-1)$, reached at $x_{n+r-1+kp}$ and $x_{n+r+kp}$, where *k* is a positive integer. The minimum output is $aK - K(r-1)$, reached at $x_{n+3r-2+kp}$ and $x_{n+3r-1+kp}$.

*1.1. Proof of Case 1.2, $C = aK + b$, where b is a positive real number such that $b < K$*

Assuming that at time-slot $n - r$ we have $x_{n-r} = aK - rK$, such that $x_{n-r}$ satisfies conditions of property (i), the output sequence increases by *K* each time-slot. Property (i) hols for $x_{n+1} = (a+1)K$ but not for $x_{n+2}$, therefore the sequence increases until reaching its maximum value $x_{n+r} = aK + rK$. See that $x_{n+r}$ satisfies (ii), therefore the sequence starts to diminish by *K* each time-slot. In $x_{n+2r} = aK$ the property (ii) applies, but not in $x_{n+2r+1}$, therefore the sequence diminishes until a minimum value at $x_{n+3r-2} = aK - K(r-1)$. Because $x_{n+3r-1}$ satisfies conditions in (i), the situation is analogous to Case 1.1, therefore the sequence restart the cycle with $4r - 2$. The maximum output value within the cycle is $aK + rK$, reached at $x_{n+r+kp}$, where *k* is a positive integer. The minimum output is $aK - K(r-1)$, reached at $x_{n+3r-1+kp}$.

*2. Proof of Proposition 2*

CA is described by the following difference equation:

$$x_n = x_{n-1} + K \cdot \left(U\left(x_{n-r} - u_{n-m} - w/2\right) - U\left(u_{n-m} - x_{n-r} - w/2\right)\right).$$

Considering an input signal $u_n = C$ for $n \geq 0$, from the time-slot when $u_{n-m} = C$, we can express CA with the following equation:

$$x_n = x_{n-1} + K \cdot f\left(x_{n-r}\right),$$

with

$$f(x) = \begin{cases} 1 & if \; x < C - w/2 \\ 0 & if \; x \in S_w \\ -1 & if \; x > C + w/2 \end{cases}$$

where $S_w$ is the set of values of the output sequence $x_n$ such that $C - w/2 \leq x_n \leq C + w/2$.

From the above equation, we can directly derive the following useful properties of the output sequence:

(i) $x_{n+m} = x_n + Km \quad for\ 0 \le m < r, \quad if\ x_{n-i} < C - w/2 \quad for\ 0 < i < r$

(ii) $x_{n+m} = x_n - Km \quad for\ 0 \le m < r, \quad if\ x_{n-i} > C + w/2 \quad for\ 0 < i < r$

(iii) $x_n = x_{n-1} \quad if\ x_{n-r} \in S_w$

*2.1. Proof of the condition for oscillation*

*2.1.1. Case 2.1.* $C - w/2 = aK$

Assuming that at time-slot $n - r$ the sequence value is $x_{n-r} = aK - rK$, satisfying conditions of property (i), the output sequence increases by *K* each time-slot. Property (i) hols for $x_n = aK$ but not for $x_{n+1}$, therefore the sequence increases until reaching its maximum value $x_{n+r-1} = (a + r - 1)K$. The sequence oscillates if this maximum value is higher than $C + w/2$ because, if so, a value $k \le r - 1$ exists such that $x_{n+r+k-1}$ satisfies (ii) and the sequence will start to decrease by one *K* each time-slot. The proof that the subsequent behaviour is cyclic is given below in subsection 2.2.1, which completes this proof. Therefore, given that $(a + r - 1)K = C - w/2 + (r - 1)K > C + w/2$, the condition for the sequence to oscillate is $w < (r - 1)K$.

*2.1.2. Case 2.2.* $C - w/2 = aK + b$*, where b is a positive real number such that* $b < K$

Assuming that at time-slot $n - r$ the sequence value is $x_{n-r} = aK - rK$, satisfying conditions of property (i), the output sequence increases by *K* each time-slot. Property (i) hols for $x_{n+1} = (a + 1)K$ but not for $x_{n+2}$, therefore the sequence increases until reaching its maximum value $x_{n+r} = (a + r)K$. The sequence oscillates if this maximum value is higher than $C + w/2$ because if so, a value $k \le r - 1$ exists such that $x_{n+r+k-1}$ satisfies (ii) and the sequence will start to decrease by *K* each time-slot. The proof that the subsequent behaviour is cyclic is given in below in subsection 2.2.2, which completes this proof. Therefore, given that $(a + r)K = C - w/2 - b + rK > C + w/2$, the condition for the sequence to oscillate is $w < rK - b$.

*2.2. Proof of the period length and oscillation amplitude expressions*

*2.2.1. Case 2.1.* $C - w/2 = aK$

Assuming that at time-slot $n - r$ the sequence value is $x_{n-r} = aK - rK$, satisfying conditions of property (i) and provided $w < (r - 1)K$ holds, the sequence then increases by *K* each time-slot until reaching its maximum value $x_{n-r-1} = (a + r - 1) \cdot K$. Because the subset $\{x_n, \ldots, x_{n+q}\} \in S_w$, the condition in (iii) holds at every value of the subset $\{x_{n+r}, \ldots, x_{n+r+q}\}$, where each element is thus equal to $(a + r - 1)K$. Conditions of property (ii) are satisfied in $x_{n+r+q}$, therefore the sequence diminish by *K* each time-slot.

Because $x_{n+2r+q-1} = aK$, then $x_{n+2r-1} = a \cdot K + qK$. Conditions in (ii) hold at $x_{n+2r-1}$ but not at $x_{n+2r}$, therefore the sequence decreases until $x_{n+3r-2} = a \cdot K - (r - q - 1)K$. Given that $\{x_{n+2r-1}, \ldots, x_{n+2r+q-1}\} \in S_w$, the condition in (iii) is satisfied by every value of the subset $\{x_{n+3r-1}, \ldots, x_{n+3r+q-1}\}$, where each element is thus equal to $aK - (r - q - 1)K$.

The conditions in (ii) hold at $x_{n+3r+q-1}$ because $x_{n+2r+q} = (a-1)K$, therefore the sequence starts to increase, until reaching $x_{n+4r} = aK - (r-q-1)K + (r-q+1)K = (a+2)K$ where concur the same conditions given at $x_{n+2}$, therefore the length of the cycle is $p = n + 4r - (n+2) = 4r - 2$.

*2.2.2. Case 2.2.* $C - w/2 = aK + b$, where *b* is a positive real number such that $b < K$

Assuming that at time-slot $n - r$ the sequence value is $x_{n-r} = aK - rK$, satisfying conditions of property (i), and provided $w < rK - b$ holds, the sequence then increases by *K* each time-slot until reaching its maximum value $x_{n+r} = (a+r)K$. Because the subset $\{x_n, \ldots, x_{n+q}\} \in S_w$, the condition in (iii) holds at every value of the subset $\{x_{n+r}, \ldots, x_{n+r+q}\}$, where each element is thus equal to $(a+r-1)K$. Conditions of property (ii) are satisfied in $x_{n+r+q}$, therefore the sequence diminish by *K* each time-slot. Because $x_{n+2r+q} = aK$, then $x_{n+2r} = aK + qK$. Conditions in (ii) holds at $x_{n+2r-1}$ but not at $x_{n+2r}$, therefore the sequence decreases until $x_{n+3r-2} = aK - (r-q-1)K$. Given that $\{x_{n+2r}, \ldots, x_{n+2r+q-1}\} \in S_w$, the condition in (iii) is satisfied by every value of the subset $\{x_{n+3r}, \ldots, x_{n+3r+q-1}\}$, where each element is thus equal to $aK - (r-j-1)K$. The conditions in (ii) hold at $x_{n+3r+q-1}$ because $x_{n+2r+q} = (a-1)K$, therefore the same conditions given in Case 2.1 concur and therefore the cycle length is $p = n + 4r - (n+2) = 4r - 2$.

*3. Proof of Proposition 3*

Applying the definition of the input signal $u_n$, it is straightforward that $u_{n-m} = C$ for $n \geq m$ and $u_{n-m} = 0$ otherwise. Therefore, $x_m = 0$ is the first value satisfying condition (i) of Proposition 1 for Case 1 and condition (i) of Proposition 2 for Case 2. In consequence, the output value *hK* appears at time slot *m* + *h* since $x_{m+h} = x_m + hK = hK$.

## 8. References


[1]. 3rd Generation Partnership Project; Technical Specification Group RAN, "Synchronisation in UTRAN Stage 2, version 7.2.0, 2006-12," 3GPP TS 25.402, Dec. 2006.

[2]. 3rd Generation Partnership Project; Technical Specification Group RAN, "Iub/Iur Interface User Plane Protocol for DCH Data Streams, version 7.3.0, 2006-12," 3GPP TS 25.427, Dec 2006.

[3]. 3rd Generation Partnership Project; Technical Specification Group RAN, "UTRAN Iub Interface User Plane Protocols for Common Transport Channel data streams, version 7.3.0, 2006-12," 3GPP TS 25.435, Dec. 2006.

[4]. J. Banister et al., "Convergence Technologies for 3G Networks IP, UMTS, EGPRS and ATM", John Wiley and Sons, 2004.

[5]. M. Sagfors et al., "Transmission Offset Adaptation in UTRAN," in Proc. IEEE VTC, vol. 7, pp. 5240- 44, Sept. 2004.

[6]. Tian Bu, Mun Choon Chan, and Ram Ramjee, "Connectivity, Performance, and Resiliency of IP-Based CDMA Radio Access Networks", IEEE Trans. on Mobile Computing, vol. 5, no. 8, 2006.

[7]. 3rd Generation Partnership Project; Technical Specification Group RAN, “Delay Budget within the Access Stratum, version 4.0.0, 2004-4,” 3GPP TR 25.853, Apr. 2004.
[8]. S. Abraham et al., “Effect of Timing Adjust Algorithms on Iub Link Capacity for Voice Traffic in W-CDMA Systems,” in Proc. IEEE VTC, vol. 1, pp. 311- 15, Sept. 2002.
[9]. C. Saraydar et al., “Impact of Rate Control on the Capacity of an Iub Link: Single Service Case,” in Proc. IEEE ICC, vol. 2, pp. 954 – 58, May 2003.
[10]. C. Saraydar et al., “Impact of Rate Control on the Capacity of an Iub Link: Multiple Service Case,” in Proc. IEEE WCNC 2003, vol. 2, pp. 1418 – 23, Mar. 2003.
[11]. M. Sagfors et al., “Overload Control of Best-Effort Traffic in the UTRAN Transport Network,” in Proc. IEEE VTC, vol. 1, pp. 456 – 60, May. 2006.
[12]. X. Pérez-Costa et al., “Analysis of Performance Issues in an IP-based UMTS Radio Access Network,” in Proc. ACM MSWiM, pp. 318 – 22, Oct. 2005.
[13]. K. Ogata, “Discrete Time Control Systems,” Prentice Hall Inc., 1994.
[14]. A. Varga (Jan. 2007). OMNeT++ Distrete Event Simulation System. [Online]. Available: http://www.omnetpp.org.
[15]. 3GPP; Technical Specification Group RAN.: IP Transport in UTRAN, 5.4.0. 3GPP TS 25.933 (2003)