UNIVERSITAT DE VIC - UNIVERSITAT CENTRAL DE CATALUNYA
(UVIC-UCC)

Algorismes per a la reconstrucció de ràfegues de dades perdudes del nivell d'aigua del dipòsit principal de la ciutat de Vic.

Tesi per a optar al Doctorat en Ciències Experimentals i Tecnologia amb menció de Doctorat Industrial

Arnau Martí Sarri

Codi 0000-0001-8886-2139

Directors:
Pere Martí Puig
Moisès Serra Serra

Octubre 2019

Agraïments

Moltes gràcies als companys de feina d'Aigües de Vic per ajudar-me en tot allò que fos de menester, especialment al Manu per posar-me totes les facilitats possibles. Moltes gràcies també al Pere i al Moisès, com a directors de tesi, per la seva assistència i consells. Finalment un especial agraïment a la família i amics, sobretot a la Mireia, que m'han animat en els moments més difícils.

Dedicatòria

Aquesta tesi està dedicada a tots aquells que m'han donat suport, especialment durant la seva la recta final.

RESUM

La majoria d'empreses que subministren aigua potable utilitzen sistemes SCADA (Supervisory Control And Data Acquisition) per controlar, supervisar i adquirir dades de la xarxa de distribució d'aigua. Aquests sistemes capturen una gran quantitat d'informació amb diferents tipus de sensors que s'emmagatzema en bases de dades per tenir registres històrics. Per utilitzar aquests historials amb garanties cal verificar-ne la seva integritat i, en alguns casos, restaurar les dades que falten, les dades malmeses o les dades que han de ser descartades a causa d'algun error. Aquesta tesi pretén desenvolupar algorismes que permetin reconstruir conjunts de dades perdudes de forma consecutiva a través d'una metodologia basada en tensors. El fet que les dades es perdin en ràfegues és comú i representa el pitjor escenari que es pot trobar, ja que a partir de certa longitud, els mètodes clàssics comencen a fallar. Per demostrar que els algorismes que es proposen funcionen correctament i superen els algorismes clàssics (no basats en tensors), es realitzen simulacions, eliminant aleatòriament conjunts de dades consecutives de l'historial i comparant les sortides proporcionades pels algorismes amb les dades originals que s'havien eliminat. Per fer-ho es disposa de dades reals registrades pel sensor de nivell d'aigua de l'embassament principal de la ciutat de Vic situat a Castell d'en Planes. A part de comparar els resultats de les diferents solucions que es proposen amb alguns mètodes clàssics, també es comparen amb una proposta de tensors existents, el mètode conegut com a "CP-Wopt", obtenint millors resultats.

ABSTRACT

Most companies that supply drinking water use SCADA (Supervisory Control And Data Acquisition) systems to control, monitor and acquire data from the water distribution network. These systems capture a large amount of information from different types of sensors that is stored in databases in order to have historical records. To use these historical records with guarantees it is necessary to verify their integrity and in some cases to restore the missing data, the damaged data or the data that must be discarded due to some error. This thesis aims to develop algorithms that can reconstruct bursts of data lost through a methodology based on tensors. The fact that data is lost in bursts is common and represents the worst case scenario which can be found, since from a certain length, the classic methods start to fail. To demonstrate that the proposed algorithms work correctly and surpass the classical algorithms (not based on tensors), simulations are performed, randomly erasing bursts of lost data in the historical records and comparing the outputs provided by the algorithms with the original data previously erased. To do that, it is used real data recorded by the water level sensor of the main reservoir in the city of Vic. Apart from comparing the results of the different proposed solutions with the classical methods, they are also compared with an existing tensor method known as "CP-Wopt", obtaining better results.

TAULA DE CONTINGUT

ÍNDEX DE FIGURES

Capítol 1

INTRODUCCIÓ

Aquest projecte sorgeix de la col·laboració entre la Universitat de Vic (UVIC) i l'empresa Aigües de Vic S.A. (AVSA) a través d'un doctorat industrial. També amb el suport de l'empresa Abastaments, Tractaments i Control d'Aigües (ATCA), una filial de la primera.

AVSA és l'empresa que gestiona la distribució d'aigua del municipi de Vic. ATCA gestiona l'Estació de Tractament d'Aigua Potable (ETAP), que recull i potabilitza aigua del riu Ter, i la distribució d'aigua als municipis de Gurb, Santa Cecília de Voltregà, Santa Eulàlia de Riuprimer i Muntanyola. Actualment l'ETAP de Miralter té una capacitat de 312,79 l/segon i 204 km de llargada, i hi ha connectats uns 21.600 abonats.

El problema que Aigües de Vic es va plantejar era que el seu sistema de Supervisió Control i Adquisició de Dades (SCADA) necessitava ser actualitzat. El sistema que tenen va ser instal·lat per una empresa externa, de la qual es depèn per incorporar millores, modificacions o en general per fer créixer el sistema, cosa que fa que el seu potencial no s'hagi aprofitat al màxim. El sistema permet visualitzar informació pràcticament en temps real, en concret amb una periodicitat de 5 minuts, però no permet realitzar un control remot dels dispositius. Això significa, per exemple, que un operari pot detectar la necessitat d'augmentar el cabal de la bomba d'aigua a l'entrada de la ETAP, gràcies a les lectures que pot fer, pràcticament en temps real, dels cabals de consum i dels nivells dels dipòsits de reserva d'aigua. L'operari, però, no pot realitzar l'acció corresponent de forma remota des de l'ordinador, ha d'anar des de l'oficina de la ETAP a les instal·lacions properes al riu Ter, on es troben les bombes de captació, i modificar manualment la freqüència del variador associat a cada bomba, segons convingui. L'inconvenient és que no pot visualitzar el cabal de la bomba assolit fins que torna a ser davant de la pantalla de l'ordinador de l'SCADA. Per tot això AVSA va decidir-se a renovar el sistema i després de valorar vàries opcions, va optar per la tecnologia de Rockwell Automation, una empresa amb prestigi reconegut.

Al realitzar la renovació de tot el sistema SCADA un dels punts a tenir en compte va ser l'aprofitament de totes les dades acumulades, cosa que implica

una migració de dades dels sistema SCADA antic al nou. Per tal de no omplir les noves bases de dades que es generarien en l'SCADA renovat amb informació innecessària, incompleta o malmesa, cal assegurar-se de quina és la informació realment rellevant i útil que cal exportar, i si la seva integritat és la adequada per poder-ne treure profit [1, 2].

Una de les dades que es va considerar important recuperar va ser el nivell del dipòsit d'aigua de Castell d'en Planes, un dipòsit d'aigua que serveix de reserva d'aigua potable a la ciutat de Vic. Les seves variacions de nivell estan directament relacionades amb el consum d'aigua de la ciutat, per tant, es podrien fer servir per trobar patrons de consum o per preveure moments crítics en la reserva d'aigua de Vic. A la ciutat de Vic hi viuen 45.040 persones (dades de 2018) en una àrea de 30.6 km^2 i és molt important que aquest dipòsit mai corri perill de buidar-se.

Analitzant el senyal d'aquest sensor es van detectar un gran número de dades perdudes. El comportament d'aquest sensor de nivell, per sort, permet una restauració molt fiable i senzilla de dades perdudes esporàdicament, degut a que el nivell d'aquest dipòsit evoluciona lentament en relació a la freqüència de mostreig del sistema SCADA i a la resolució del sensor. És a dir, tenint en compte que el sensor té una resolució de l'1 % respecte a la capacitat total del dipòsit (16.000 m^2), que la freqüència de mostreig és de 5 minuts, i que les condicions de consum d'aigua, o sigui d'entrada i sortida d'aigua del dipòsit, fan que sigui impossible buidar o omplir més de l'1 % del dipòsit en 5 minuts, és impossible que entre una mostra i la següent hi hagi una diferència major al 1 %. Això fa que la pèrdua d'una mostra concreta, enmig d'una serie determinada de mostres, sigui relativament fàcil d'estimar amb un alt percentatge d'efectivitat. Si augmenta el nombre de mostres perdudes, llavors la capacitat de recuperar les dades originals, dependrà de com es distribueixin aquestes mostres perdudes. Si és perden de forma intercalada, les dades es recuperen fàcilment i amb efectivitat, però si les mostres es perden de forma consecutiva, llavors el problema es més complicat. En casos on la pèrdua de dades es deguda a la cal·libració d'un sensor, a fallades puntuals de comunicació entre dispositius, al re-inici d'un PLC ("Programmable Logic Controller.º Controlador Lògic Programable) o del propi servidor SCADA la pèrdua de dades sol ser de la primera forma i per tant fàcil de restaurar eficaçment mitjançant tècniques o mètodes lineals [3–7]. D'altre banda en els casos de talls importants de comunicació en algun punt de la xarxa del sistema SCADA o fallades greu

d'algun PLC o d'algun sensor, llavors la pèrdua de dades sol ser en forma de ràfega de mostres perdudes, més complicades de restaurar. És a dir, quan per exemple la reparació d'un problema de comunicacions amb l'SCADA s'allarga, es produeix la pèrdua d'un llarg conjunt de mostres consecutives, difícil de recuperar amb les tècniques o mètodes lineals habituals.

1.1. Objectiu

Aquesta tesi es centra en l'estudi i desenvolupament de tècniques de recuperació de dades perdudes en forma de ràfega. La hipòtesi principal és que els tensors tenen el potencial de capturar patrons o certes periodicitats que un senyal segueix, encara que siguin difícils de veure a simple vista, de forma que es poden utilitzar per realitzar la restauració de dades perdudes amb més eficàcia que els mètodes lineals convencionals. Sobretot quan les dades que falten i que cal recuperar són mostres que s'han perdut de forma consecutiva, on els mètodes convencionals solen augmentar considerablement el seu biaix i l'àlgebra tensorial permet assolir millors resultats.

Per al desenvolupament de la tesi s'utilitzen dades de la xarxa de distribució d'aigua, no obstant, és un problema comú en molts altres camps. Això significa que part de les tècniques descrites es poden exportar a altres àrees de coneixement. El fet de disposar d'una base de dades prou completa d'un cas real té un valor incalculable que ens permet provar els mètodes desenvolupats amb simulacions basades en dades reals. Com s'ha comentat, es focalitza en la situació més difícil de resoldre, el cas de la pèrdua de dades consecutives o a ràfegues. La causa en aquest cas pot ser un error o fallada del sensor que requereixi tornar a calibrar-lo, un fallada de l'enllaç de comunicació entre el sensor i el PLC o entre el PLC i el servidor, o directament un fallada del sistema SCADA o de la seva unitat d'emmagatzematge de dades. En aplicacions pràctiques cal tenir molt en compte la verificació i recuperació de dades. La diferència entre introduir dades, més o menys fiables, a la primera etapa de la cadena de processament, pot ser un factor molt important, ja que aquests valors s'aniran propagant a la resta de les etapes i poden acabar tenint un gran impacte en el resultat final.

1.2. Estructura de la memòria

La memòria de la tesi s'organitza en capítols seguint la lògica dels articles presentats a revistes amb factor d'impacte. L'objectiu inicial era presentar la tesi per compendi de publicacions. En el moment d'escriure-la encara resta pendent la finalització de la revisió de la tercera publicació que es requereix a la UVic-UCC. A causa d'aquest contratemps es presenta la feina feta en format de capítols i s'adjunten els articles presentats i el "pre-print" d'aquest últim que està en procés de revisió. La tesi té un fil conductor que coincideix amb el seu desenvolupament cronològic. Els capítols 2 i 3 serveixen per introduir-se en el context de la tesi. Els capítols 4,5 i 6 presenten una introducció inicial i dos apartats finals amb resultats i conclusions parcials, per fer-los més entenedors.

Al capítol 2 s'explica quins tipus de dades perdudes es poden trobar o classificar i quines són les metodologies utilitzades actualment per resoldre aquest problema, posant com a exemple el cas d'estudi.

Al capítol 3 es descriu la metodologia emprada. S'explica com s'han obtingut les dades que s'utilitzen en les simulacions i quines eines es fan servir. Es mostra el procés de verificació de les dades que s'ha dut a terme abans de fer les simulacions. També es detalla la forma d'avaluar els algorismes de reconstrucció de ràfegues proposats i en general en quines condicions es realitzen els experiments.

Al capítol 4 s'expliquen els mètodes que fan servir els operaris d'Aigües de Vic quan necessiten realitzar l'estimació de dades perdudes. En una primera fase per investigar com proporcionar una eina útil de restauració de dades, es planteja l'opció d'utilitzar un mètode lineal que permeti restaurar la informació a través dels possibles patrons o periodicitats intrínseques a la mesura realitzada pel sensor. Al capítol 5 s'introdueix el concepte dels tensors i s'investiguen alguns mètodes de restauració de dades que fan servir aquesta eina [8, 9]. Els tensors permeten organitzar les dades a diferents nivells. A partir d'introduir les dades en un tensor i organitzar-les convenientment d'una determinada manera, es pot realitzar una descomposició tensorial [10–12] que permet capturar patrons relacionats amb l'organització de les dades, que es poden fer servir per reconstruir dades perdudes. Es desenvolupa una metodologia basada en la descomposició tensorial per recuperar mostres perdudes en forma de ràfega, que fa servir aquest procés combinat amb un mètode lineal previ i un algoritme posterior dissenyat per mantenir la continuïtat del senyal.

Al capítol 6 s'aprofundeix en la metodologia desenvolupada en el capítol 5 per millorar-ne el rendiment. Es detecta un problema amb la continuïtat del senyal restaurat relacionat amb la posició de la ràfega de dades perdudes dins el tensor. Per evitar-lo, millorant el rendiment de l'algoritme, es dissenya una reorganització del tensor segons la posició de la ràfega, que permet minimitzar l'efecte de les discontinuïtats. Es realitzen un parell d'observacions més referents als resultats de les descomposicions tensorials que permeten polir l'algorisme. En concret un procés de suavitzat del senyal i l'aplicació d'una doble descomposició tensorial que es complementen per assolir resultats encara més acurats en l'estimació de les mostres perdudes.

Per acabar la tesi al capítol 7 es recopilen les conclusions globals i es proposen futures millores i experiments.

1.3. Publicacions relacionades amb la tesi

A continuació es llisten els articles publicats o pendents de revisió, relacionant-los amb els capítols corresponents i de forma cronològica.

El juny de 2018 es va presentar un article al congrés "9th International Congress on Environmental Modelling and Software: Modelling for Sustainable Food-Energy-Water Systems" (IEMSS 2018), celebrat a Ford Collins (Colorado, US), apèndix A. S'hi va assistir per fer una presentació de l'article, que fa referència al capítol 4 de la tesi, on s'explica l'algoritme per a la reconstrucció de dades de la secció 4.3, que a més es compara amb els mètodes usats pels operaris d'Aigues de Vic explicats a les seccions 4.1 i 4.2. En el mètode lineal de reconstrucció de dades desenvolupat s'adapta el predictor de Wiener [13–17] obtenint un algorisme que permet reconstruir ràfegues de dades perdudes a través de l'historial de dades del senyal a restaurar, mitjançant un conjunt de dades anteriors i posteriors a la ràfega perduda. S'utilitzen dos versions del filtre de Wiener per obtenir una reconstrucció de la ràfega de mostres perdudes, que no perd de la continuïtat del senyal en els extrems de la ràfega.

El març de 2019 es va enviar l'article de l'apèndix B a la revista"Water", que després d'una revisió, va publicar el maig de 2019 a la secció "Urban Water Management". En aquest article s'aprofundeix sobre l'anomenat mètode "FIR"de la secció 4.3 presentat al congrés IEMSS 2018. També s'investiga com els tensors, explicats al capítol 5, poden ajudar a millorar els resultat obtinguts

amb els mètodes lineals del capítol 4, a l'hora de realitzar la reconstrucció de les dades perdudes en forma de ràfega, [8, 9, 18]. I finalment es proposa un mètode basat en tensors per a la reconstrucció de ràfegues de dades perdudes.

El maig de 2019 es va presentar l'article publicat al 22è Congrés Internacional de l'Associació Catalana d'Intel·ligència Artificial (CCIA 2019), apèndix C. En aquest article es presenta una de les millores proposades al capítol 6 pel mètode de de reconstrucció de ràfegues basat en tensors de l'article anterior (explicat al capítol 5). S'exposa una nova manera d'ordenar les dades en funció de l'hora del dia i el dia de la setmana en que es produeix la ràfega de dades que cal restaurar.

Finalment el setembre de 2019 es va enviar l'article de l'apèndix D a la revista "Water" (secció "Urban Water Management") que està pendent de revisió. En aquest article s'expliquen totes les millores proposades al capítol 6 per la metodologia de reconstrucció de dades del capítol 5, que permeten assolir resultats encara més acurats en l'estimació de les mostres perdudes.

Capítol 2

ESTAT DE L'ART

Els sistemes SCADA reben, transmeten i guarden una gran quantitat de dades provinent de múltiples sensors i dispositius. Si en algun punt d'aquest procés es produeix una fallada o una interrupció, probablement les dades es perdin. És a dir una dada es pot perdre si la mesura no es realitza per algun problema del sensor, si no es pot comunicar correctament la lectura al servidor central o si el sistema falla en el moment d'emmagatzemar la dada. A més, les dades guardades han de ser validades abans de poder-se fer servir per analitzar l'estat del sistema i controlar-lo, fer previsions, evitar avaries, etc. Si no es verifica la integritat de les dades els models o elements que les fan servir probablement estaran distorsionats i la informació que proporcionin no serà fiable [2, 19]. Al realitzar aquests procediments de verificació les dades incorrectes o incoherents, les dades detectades com a dolentes solen ser descartades i es poden tractar com a dades perdudes. El problema d'haver de gestionar com es tracten les dades perdudes, no obstant, és un problema general que trobem en tots els sistemes que tracten dades capturades per sensors durant llargs intervals de temps [20, 21].

Les dades es poden perdre per diferents causes o de diferents maneres. De forma general es poden definir tres grups importants anomenats "Missing At Random" (MAR), "Missing Completely At Random" (MCAR), i "Missing Not At Random" (MNAR), [22]. Per explicar aquesta classificació es considera un historial de dades X_{hist} que es pot dividir en dos grups dades, el format per les mostres mesurades i verificades X_{mes} i el format per les dades perdudes o descartades X_{perd}, de manera que $X_{hist} = (X_{mes}, X_{perd})$. Tenint en compte això, el grup anomenat MAR fa referència a aquells casos en que la probabilitat que es perdin les dades només depèn de les dades mesurades X_{mes}, i no de les dades perdudes X_{perd}. El grup MCAR, en canvi, inclouria els casos en que les dades es perden de forma completament aleatòria i no depenen ni de les dades perdudes X_{perd}, ni de les mesurades X_{mes}. Finalment el tercer grup, MNAR, es refereix als casos en que les dades que es perden depenen de les pròpies dades perdudes . Per posar l'exemple del cas d'estudi concret d'aquesta tesi, el sensor de nivell d'aigua del dipòsit de Castell d'en Planes, es pot assumir que

estem en el grup MCAR, ja que les pèrdues de dades dels historials d'aquest sensor no depenen del nivell d'aigua del dipòsit ni de cap dada disponible.

Per definir d'una forma més formal aquest conceptes, es considera R una matriu de la mateixa mida que X_{hist}, que indica amb "1" les dades mesurades i amb "0" les perdudes. Es considera ξ un indicador dels paràmetres desconeguts relacionats amb la pèrdua de dades i es defineix la distribució de probabilitat de les dades perdudes com $P(R/X_{hist},\xi) = P(R/X_{mes},X_{perd},\xi)$, [23]. Llavors es concreta l'equació per a cada un dels grups de la següent manera.

Per al cas MAR es compleix que:

$$P(R = 0/X_{mes},X_{perd},\xi) = P(R = 0/X_{mes},\xi).$$

Per al cas MCAR:

$$P(R = 0/X_{mes},X_{perd},\xi) = P(R = 0/\xi).$$

I de fet per al grup MNAR no es pot simplificar l'expressió:

$$P(R = 0/X_{mes},X_{perd},\xi).$$

Hi ha varis problemes derivats dels historials de dades incompletes. Un d'ells és la pèrdua d'informació, que redueix l'eficiència del sistema. Aquest problema és difícil de solucionar ja que per no perdre informació s'hauria de millorar el sistema de forma que no es perdessin dades, es pot optar per sistemes redundants, però al final és inherent en tots els casos la possibilitat de fallar, ja que cap sistema és perfecte. Un altre problema és la complicació que afegeix a l'hora de realitzar el processament o anàlisis de les dades, sobretot amb les tècniques que no permeten historials de dades irregulars. Aquest és un problema fàcil de veure i contra el qual es pot intervenir de vàries maneres, ja sigui adaptant les tècniques per treballar amb historials irregulars, o completant els historials mitjançant la reconstrucció de les dades perdudes per poder fer servir totes les eines de processament o anàlisis de les dades clàssics. Potser aquest és el punt on es pot trobar més literatura sobre aquest tema. Finalment la pèrdua de dades suposa la introducció de biaix o error en els càlculs o resultats. Aquest últim problema segurament és el més fonamental, ja que les dades perdudes signifiquen clarament una dificultat al moment de representar correctament el que s'estigui mesurant. Es poden millorar els sistemes de càlcul i processament per tal que tinguin en compte l'error introduït per les dades perdudes, però és una tasca complicada ja que al solucionar el problema en la seva última etapa, les raons per les que no es disposa de les

dades completes i la seva influència en el resultat final, solen ser més difícils d'analitzar i entendre [24].

Per atacar el problema de la pèrdua de dades hi ha vàries opcions. Les més simples és ignorar o descartar els conjunts de dades si és possible, com en el mètode anomenat "Listwise deletion", en el que es descarten els blocs de dades amb mostres perdudes. O el mètode "Pairwise deletion" en el que s'intenten aprofitar els blocs de dades amb pèrdues a base de covariàncies i correlacions [22]. En el cas estudiat interessa recuperar el senyal complet del sensor de nivell, que permet estudiar-ne els patrons de consum, per tant aquests mètodes no són útils. Ho podrien ser en cas que l'objectiu fos, per exemple, calcular la mitja diària del nivell del dipòsit. Quan aquest mètodes introdueixen massa error, no convenen o no és possible usar-los per algun motiu, el més habitual és utilitzar mètodes de reconstrucció de dades.

En l'àmbit de la reconstrucció de dades hi ha molta informació i literatura. Per fer una divisió inicial es pot parlar de tres classificacions de mètodes de reconstrucció de dades.

El primer grup i el més més senzill seria el de variable única, bàsicament es substitueixen els valors perduts per estimacions, mantenint la mida dels historials de dades. S'utilitzen anàlisis estadístics per estimar els valors perduts amb valors plausibles. En aquests casos, s'assumeix que hi haurà un error, perquè degut a la incertesa generada per les dades perdudes, no es pot saber amb fiabilitat fins a quin punt els valors estimats s'assemblen als valors real. Un exemple n'és l'anomenat "mean imputation", que manté la mitja de la variable, però redueix la variància provocant una distorsió en la distribució de la variable. O mètodes molt semblants com serien el "Mode imputation" o el "Median imputation" que fan servir el mode i la mitjana respectivament. També trobem mètodes més complexes com per exemple els basats en la regressió del senyal, però que també provoquen distorsió en la distribució de la variable analitzada. Finalment els mètodes més elaborats d'aquest tipus són els que es fonamenten en la regressió estocàstica, que fan servir models de predicció per obtenir estimacions que conserven més fidelment la distribució de la variable.

El segon grup seria el format per mètodes de variable múltiple que són aquells en que vàries variables formen un sistema o un model. En aquests casos el més important es seleccionar correctament les variables implicades i construir

un model fidedigne. Normalment es divideixen en tres etapes. En la primera es realitzen vàries estimacions o reconstruccions de les dades a partir de mitges, covariàncies, regressions i en general paràmetres estadístics, de forma semblant als mètodes predictius comentats anteriorment. En la segona es realitza un anàlisis estadístic dels resultats, que dependrà de la quantitat d'estimacions realitzades a la primera etapa. En la tercera fase les diferents estimacions es posen en comú, combinant-les per generar el resultat final de la reconstrucció de dades [25].

El tercer grup està format per mètodes que treballen amb les funcions de probabilitat i busquen maximitzar o optimitzar algun paràmetre o error. En són exemples els mètodes "Full Information Maximum-Likelihood" (FIML) o "Expectation Maximization" (EM). Amb aquestes metodologies els valors perduts no es reemplacen sinó que es fan servir algoritmes de maximització de les expectatives per estimar els paràmetres que defineixen el model. D'aquesta forma el que s'estima són els paràmetres del model i no les dades perdudes en si. Es tracta d'utilitzar les dades disponibles per definir el model més fiable possible. Aquests mètodes donen resultat semblants als de variable múltiple, sobretot quan s'hi incorpora exactament la mateixa informació [26].

És important remarcar que els dos últims grups tenen l'objectiu de definir el model lo més correctament possible, és a dir estimar el paràmetres que el defineixen lo millor possible, tot i les dades perdudes. No tenen l'objectiu d'estimar o predir els valors concrets de les mostres perdudes. Per tant pel cas del sensor de nivell que es tracta, aquests mètodes no són adequats. Serien útils si es combinessin més variables com podrien ser els cabal d'entrada i sortida d'aigua del dipòsit. Llavors es podria, per exemple, fer càlculs de pèrdues o del balanç d'aigua. Incloure aquestes variables o paràmetres en el procés a estimar permetria generar un sistema que es podria modelar. Per tant per al cas d'estudi d'aquesta tesi s'analitzen varis mètodes corresponents al primer grup, del qual es provaran mètodes relacionats amb tots els que s'han comentat d'aquest grup. En concret es fa servir el mètode explicat a l'apartat 4.1, com a mètode semblant als anomenats "Median imputation", "Mode imputation" o "Median imputation", ja que tots ells substitueixen totes les mostres de la ràfega perduda amb un valor concret. En el mètode proposat en lloc de fer servir un paràmetre estadístic utilitza l'ultim valor rebut correctament. També es fa servir un mètode similar al de la regressió, apartat 4.2, en el que per omplir les dades perdudes es calcula el pendent de la recta que permet unir la

última dada rebuda correctament abans de la pèrdua de dades, amb la primera dada rebuda correctament després de la pèrdua de dades. A l'apartat 4.2 es proposa un mètode inspirat en els de regressió estocàstica, fent servir vàries versions del predictor de Wiener. Per últim, en el cos de la tesi, es proposa una metodologia que fa servir els tensors, una eina que permet representar el senyal vectorial del sensor de nivell amb format matricial o fins i tot amb una estructura que tingui més de dues dimensions. Amb aquesta tècnica és raonable parlar d'una metodologia situada a mig camí entre el primer grup i els segons, tot i la restricció que òbviament suposa, treballar amb una sola variable.

Capítol 3

METODOLOGIA

En aquest capítol s'explica com s'han obtingut les dades i com es realitzen els càlculs i les simulacions necessàries per als experiments proposats a les seccions 3.1 i 3.2.

Tot seguit, a la secció 3.3, es descriu el procés realitzat per descartar mostres incorrectes degut a algun error del sensor o del PLC. Al acabar aquest procés, es disposa d'un historial de mostres del senyal de nivell, amb una freqüència de 5 minuts, i amb alguns buits deguts a mostres perdudes o descartades.

Finalment es defineixen les condicions en les que es realitzaran les simulacions de ràfegues perdudes en aquesta tesi, secció 3.4.

3.1. Obtenció de les dades

Les dades que s'utilitzen en les simulacions són proporcionades per Aigües de Vic, obtingudes directament del seu SCADA que va començar a funcionar a finals de 2014. Tot i que realment fins l'octubre de 2015 no va estar operatiu per guardar historials de dades i capturar correctament les dades dels sensors més importants. Des de llavors, totes les dades s'emmagatzemen a una base de dades SQL que està al servidor central d'Aigües de Vic. Aquest servidor es comunica amb la resta d'elements de l'SCADA, bàsicament els PLCS, per guardar tota la informació. El sistema de comunicació que fa servir cada node depèn de la seva localització, els que estan a la planta de tractament d'aigua es connecten via cablejat, però els que estan fora de la planta es connecten sobretot via ràdio. L'antiguitat del sistema fa que sovint es produeixin fallades en la comunicació amb algun PLC. Actualment l'SCADA rep uns 1300 senyals diferents de la estació de tractament d'aigua potable i de la xarxa de distribució tals com mesures de pH, clor, cabal, pressió, la freqüència dels variadors de les bombes, la posició de les vàlvules, nivells de dipòsits, boies de seguretat, etc.

En les simulacions s'utilitzen les mesures del sensor de nivell de Castell d'en Planes, la reserva principal d'aigua de la ciutat de Vic.

3.2. Eina per al tractament de dades

Per a realitzar tots els càcluls, operacions i simulacions s'han fet servir el Matlab 2016 i el 2018, [27, 28]. Per poder recollir les dades es va utilitzar la llibreria "Database Toolbox"del Matlab, que ofereix les comandes extres del Matlab per connectar via SQL amb bases de dades externes. Amb el permís d'Aigües de Vic vam obtenir l'historial de mesures del sensor de nivell de Castelll d'en Planes i el vam convertir en un vector de dades fàcil de tractar.

S'utilitza el Matlab aprofitant que hi ha una llicència disponible de la universitat. Amb aquest programa es realitza tot el tractament matemàtic i el processament de senyals que es requereix per fer les simulacions.

Pel tractament complexe que demanen els tensors es fa servir la llibreria "Tensor Toolbox" i per aplicar l'algoritme CP-Wopt la llibreria "Poblano Toolbox".

A més el Matlab permet fer la representació gràfica dels resultats obtinguts que es mostren al llarg de la tesi en les figures.

3.3. Pre-processament de dades

Els sistemes SCADA permeten rebre i emmagatzemar una gran quantitat de dades provinents de diferents sensors pràcticament en temps real. De forma que a part de reaccionar al moment, permeten treballar amb els historials de dades acumulades. Però abans de poder fer servir aquestes dades, cal verificarne la seva integritat, [1, 2, 19].

Per assegurar-se que les dades utilitzades en les simulacions són fiables es realitza un procés de verificació del senyal a diferents nivells [2] (pag 175-191). Es consideren les condicions concretes del sensor de nivell del dipòsit de Castell d'en Planes per tal de detectar i poder descartar errors en les mesures emmagatzemades pel sistema SCADA. Es descarten els valors detectats com a incorrectes segons els criteris descrits a continuació.

1. Nivell temporal: verificar que la marca de temps és correcte. Es comprova que el registre a la base de dades, corresponent al moment en que es guarda cada mostra, compleix certes condicions. En el cas del sensor de nivell la freqüència de mostreig de l'SCADA és de 5 minuts. Les dates emmagatzemades en el registre temporal de la base de dades tenen el següent format:

$$AAAA/MM/DD\text{ - }hh:mm:ss$$

On cada lletra representa un digit i l'esquema anterior equival a:

$$(ANY/MES/DIA\text{ - }hora:minut:segon)$$

Les condicions que cal complir són les següents condicions:

$$AAAA = (2015, 2016, 2017, 2018, 2019),\ MM = (1, 2, 3..., 12),\ DD = (1, 2, 3..., 31)$$

$$hh = (0, 1, 2..., 23),\ mm = (0, 5, 10, ..., 55),\ ss = 0.$$

Bàsicament es verifica que la data i la hora estiguin dins del rang possible, i que en concret els minuts siguin múltiple de 5 i els segons 0. Es troben 421 mostres d'un historial de 421.533 amb marques de temps irregulars, que són descartades. Al ordenar les dades restants des de la primera a la última amb una separació de 5 minuts s'obté un bloc de 574.848 posicions, de les quals, òbviament, 153.736 són buides. A continuació es mostren alguns exemples en les figures 3.1-3.3.

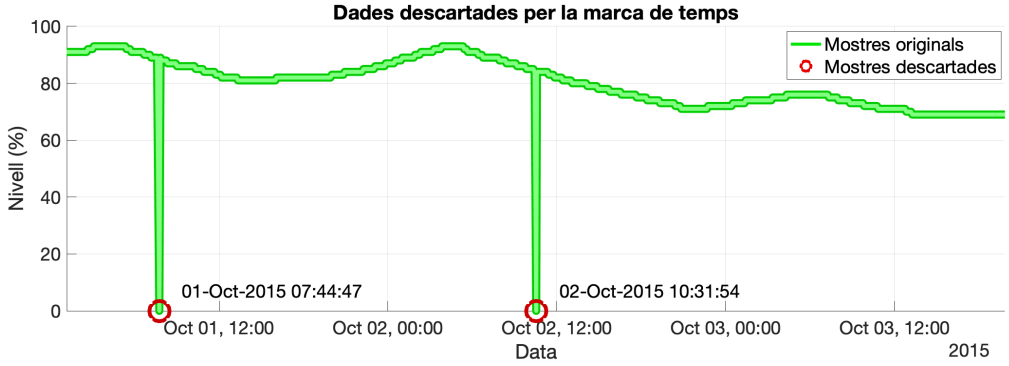

Figura 3.1: Exemple de mostres descartades degut a que el registre temporal guardat a la base de dades és incoherent. Ni els minuts compleixen la periodicitat (no són múltiple de 5), ni els segons són 0.

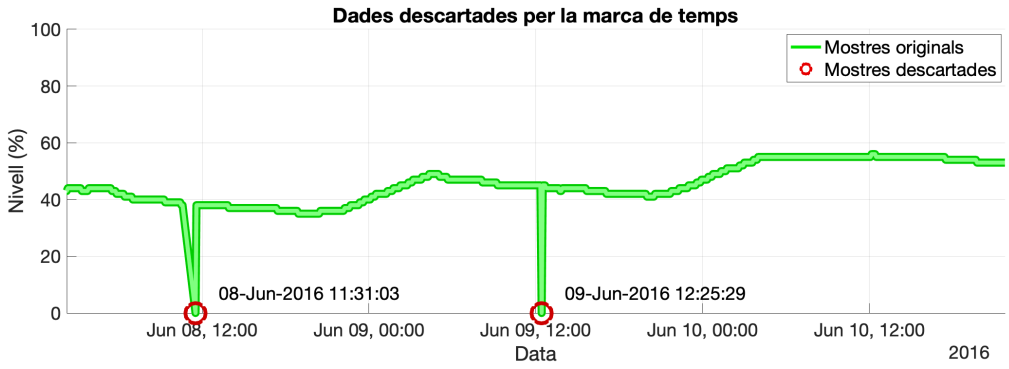

Figura 3.2: Exemple de mostres descartades degut a que el registre temporal guardat a la base de dades és incoherent. Ni els minuts compleixen la periodicitat (no són múltiple de 5), ni els segons són 0.

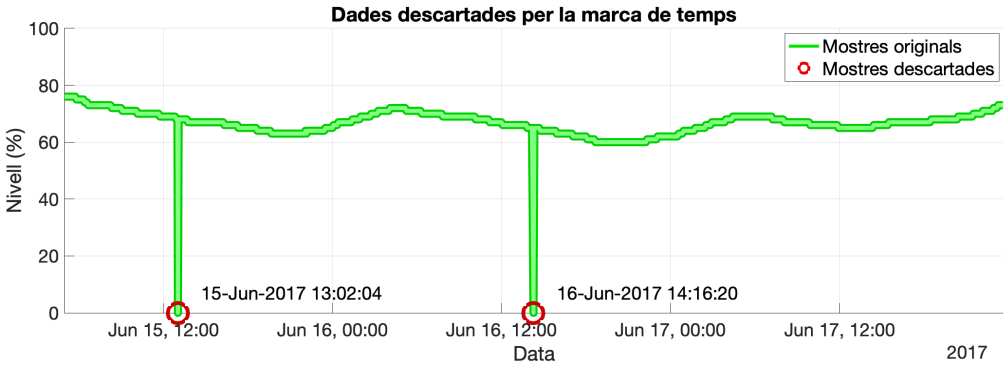

Figura 3.3: Exemple de mostres descartades degut a que el registre temporal guardat a la base de dades és incoherent. Ni els minuts compleixen la periodicitat (no són múltiple de 5), ni els segons són 0.

2. Nivell físic: verificar les mesures segons les lleis físiques i les condicions de l'entorn del sensor. En aquest cas la mesura és un percentatge que depèn de la capacitat total del dipòsit ($16.000 m^3$). Degut a la resolució de 1 % del sensor, el valor sempre és un enter. El rang de valors possibles va de 30 a 100 ja que aquest diposit mai està buit. Es descarten les dades que no compleixen aquestes condicions. Es troben 183.744 mostres inicials amb valor 0 o nul, que són descartades, ja que el dipòsit no pot estar buit. A les 391.104 posicions restants hi ha 389.494 mostres i 1.610 posicions buides. Al analitzar les dades es troben 416 mostres que no compleixen els valors mínims i màxims possibles, per tant es descarten deixant-ne 389.078 i 2026 posiciones buides. A continuació es mostren alguns exemples en les figures 3.4-3.6.

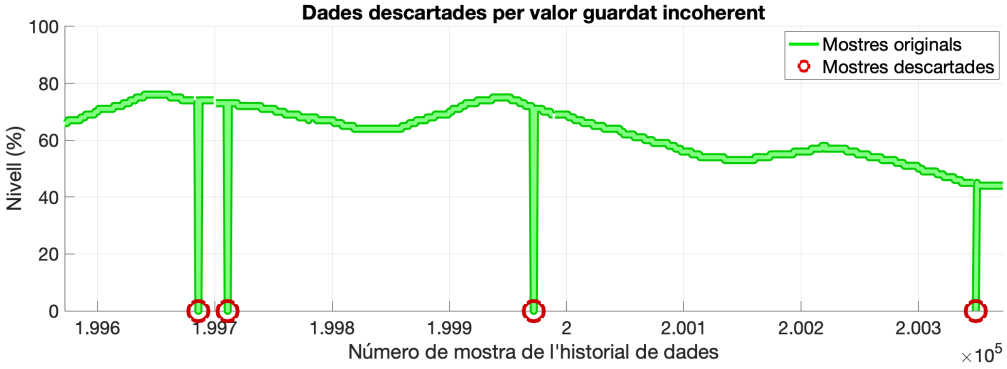

Figura 3.4: Exemple de mostres descartades degut a que els valors registrats són incoherents. En aquest cas són totes 0, que no es troba dins del rang vàlid, entre 30 i 100. Com que són dades situades esporàdicament, es fa difícil saber el motiu del registre erroni. Podria ser degut al reinici d'algun dispositiu, com ara un PLC o el propi servidor SCADA.

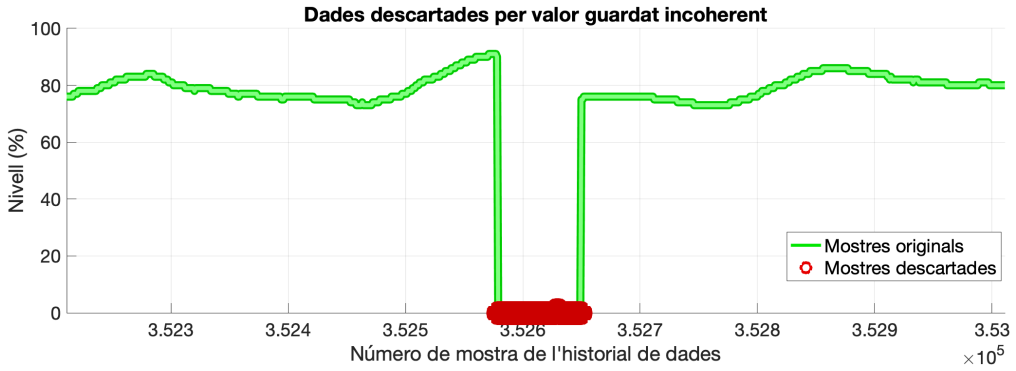

Figura 3.5: Exemple de mostres descartades degut a que els valors registrats són incoherents. En aquest cas són totes 0, que no es troba dins del rang vàlid, entre 30 i 100. Aquest cas podria ser degut a un problema amb el sensor, ja es registren molts valors seguits incorrectes.

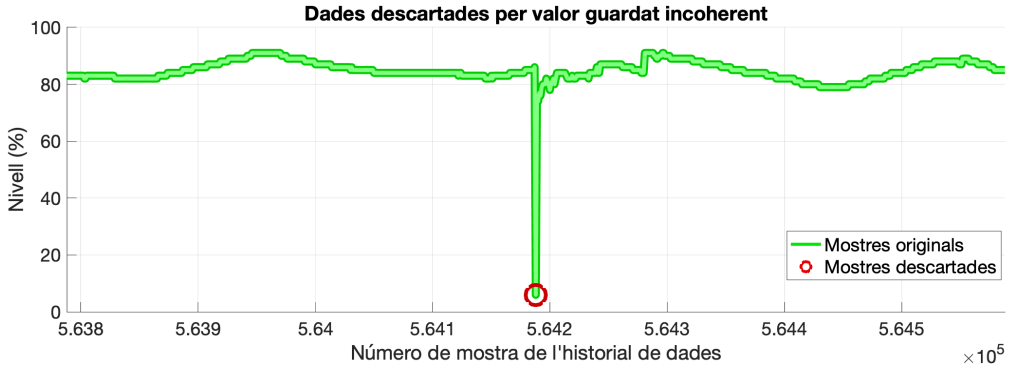

Figura 3.6: Exemple de mostres descartades degut a que els valors registrats són incoherents. En aquest cas el valor registrat no és 0, però és inferior a 30 i per tant tampoc es troba dins del rang de valors acceptats, entre 30 i 100. Pot ser un desajust del sensor ja que les mostres següents tenen mal aspecte i un temps després sembla que es re-calibra el sensor.

3. Nivell de tendència: validar el valor de cada mesura respecte les mesures anteriors. En aquest cas, degut a les condicions del bombament que permet omplir el dipòsit i a les dels consumidors que el buiden, entre una mostra i la següent com a molt hi pot haver un 1 % de diferència. La velocitat del flux d'aigua que es genera no permet més variació, ni en cas d'omplir, ni en cas de buidar, per tant es descarten les mostres que no compleixen l'increment o decrement màxim establert. Es troben 10 mostres que cal descartar degut a que no compleixen aquesta condició, deixant uns totals de 389.068 mostres i 2036 posicions buides. A continuació es mostren alguns exemples en les figures 3.7-3.9.

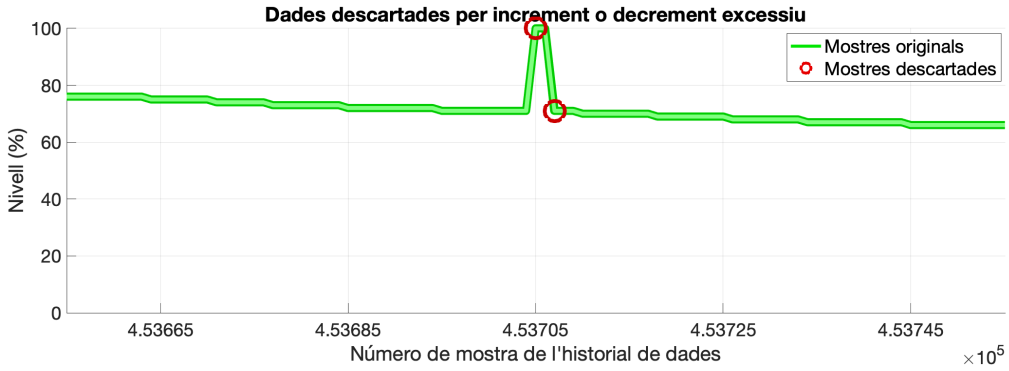

Figura 3.7: Exemple de mostres descartades perquè l'increment o decrement entre elles és major al 1 %. Sembla un error puntual en el sensor.

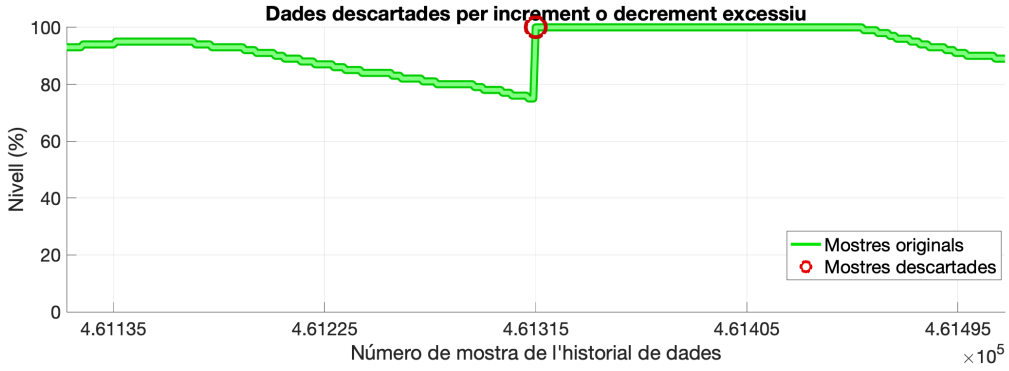

Figura 3.8: Exemple de mostres descartades perquè l'increment o decrement entre elles és major al 1 %. S'observa un desajust sobtat del sensor. Segurament va ser necessari re-calibrar el sensor.

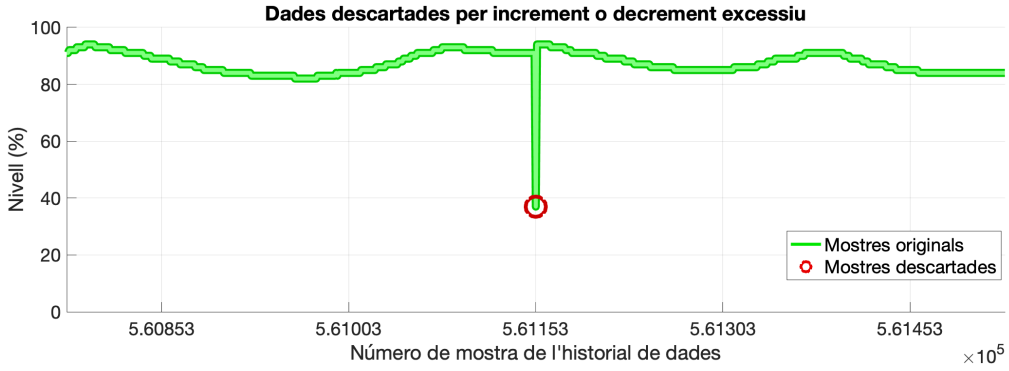

Figura 3.9: Exemple de mostres descartades perquè l'increment o decrement entre elles és major al 1 %. Sembla un error puntual del sensor.

4. Nivell d'equipament: la mesura és coherent comparada amb la resta de mesures relacionades, preses per sensors diferents. En aquest sentit es comparen les mesures de nivell amb les dels mesuradors de cabal d'entrada i sortida, ja que la diferència de cabals està directament relacionada amb el nivell del dipòsit. La idea és comprovar que la diferència de cabals és coherent amb els canvis de nivell. En les proves realitzades no es troben errors del sensor de nivell per aquest motiu.

Després d'aquests processos, s'analitzen les dades perdudes o descartades tenint en compte les que es situen consecutivament, i es calcula la mitja de la mida de les ràfegues obtenint 12,8 mostres. Però si es considera que una ràfega requereix un mínim de durada, per exemple una hora de duració (el que serien 12 mostres), i es descarten la resta de casos del recompte, la mitja puja fins a 103,4. Si es considera una ràfega mínima de 25 mostres el resultat és de 141,8.

Quan es fa referència a dades perdudes, tant poden ser aquelles que es perden degut a una fallada del sistema de comunicacions o emmagatzematge de dades, com aquelles que es descarten per tenir valors incoherents o incorrectes (per exemple degut a un sensor que falla) [29].

3.4. Simulacions

Per realitzar les simulacions s'introdueix l'historial de mostres del sensor de nivell, recollides pel sistema SCADA i emmagatzemades a la base de dades del servidor central, al programa Matlab mitjançant comandes SQL.

Abans de realitzar les simulacions, s'eliminen les setmanes de l'historial amb excessives mostres perdudes o mostres que s'han de descartar per ser incorrectes. De forma que al fer les simulacions es treballa només amb les setmanes que contenen dades fiables i verificades. En la secció 3.3 es descriu amb més detall el procediment per descartar mostres incorrectes.

Finalment, per seleccionar les setmanes de les simulacions es revisa visualment l'historial del senyal del sensor, s'observa setmana a setmana descartant aquelles setmanes que mostren algun problema, com ara falta de dades o error del sensor. A continuació es mostren alguns exemples en les figures 3.10-3.15.

Cada simulació consisteix en l'eliminació intencionada d'un seguit de mostres consecutives, que representen la ràfega de dades perduda, dins l'historial de

Figura 3.10: Exemple d'una setmana acceptada.

Figura 3.11: Exemple d'una setmana acceptada.

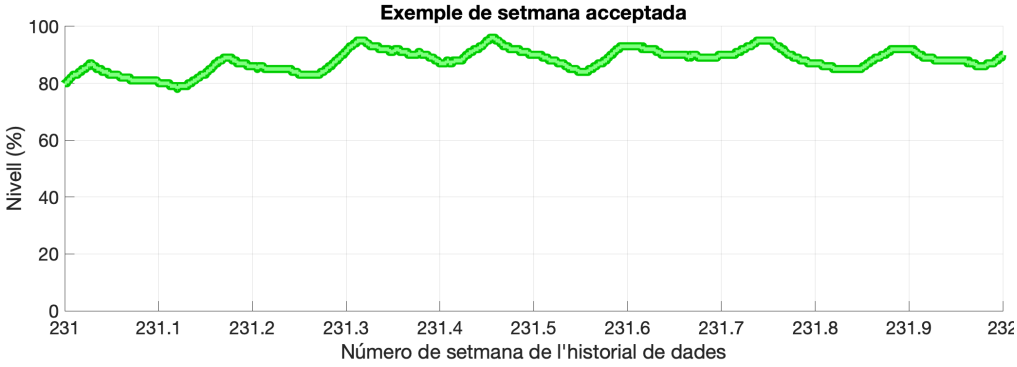

Figura 3.12: Exemple d'una setmana acceptada.

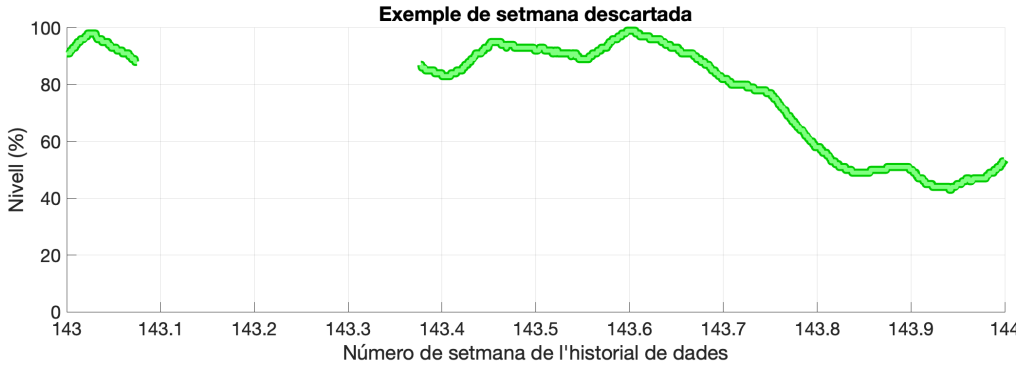

Figura 3.13: Exemple d'una setmana descartada. Es produeix un fallo greu de comunicació entre el sistema SCADA i el PLC del dipòsit de Castell d'en Planes, o entre el PLC i el sensor de nivell, ja que es perd el senyal durant més d'un dia seguit.

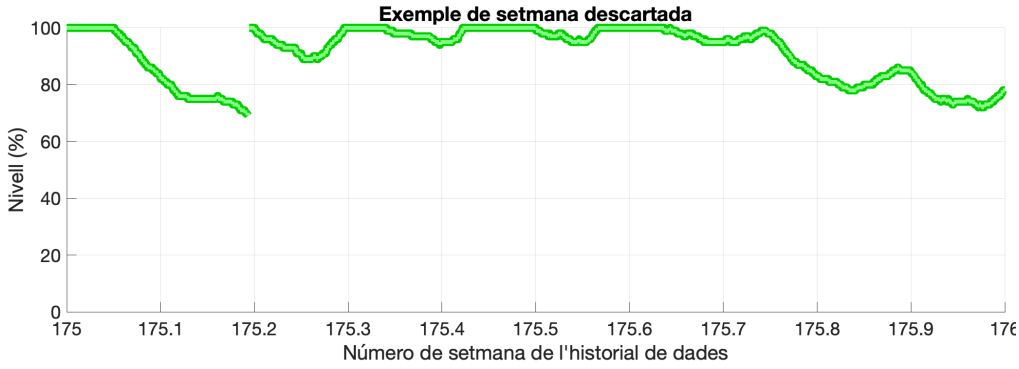

Figura 3.14: Exemple d'una setmana descartada. Es produeix un fallo de calibratge. Després d'un salt esporàdic de nivell s'observen vàries saturacions del sensor al nivell, ja que indica màxim vàries vegades seguides.

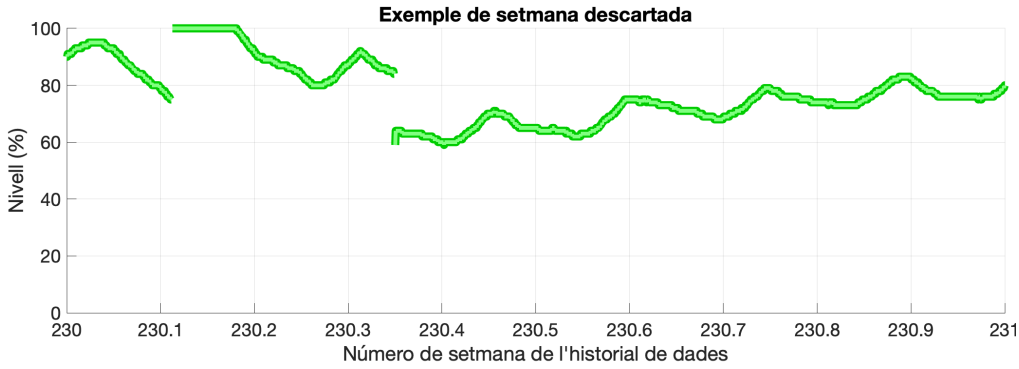

Figura 3.15: Exemple d'una setmana descartada. Es produeix un fallo de desplaçament de la lectura probablement el sensor va rebre un impacte lleu cosa que va desviar la seva lectura fins que els operaris se'n van adonar i van tornar a calibrar-lo.

dades verificat del sensor de nivell. Després s'aplica el mètode de reconstrucció de dades que es vol provar. D'aquesta forma es pot fer servir l'error quadràtic mig ("Mean Squared Error", MSE) com a mesura de comparació, ja que es disposa dels valors restaurats amb el mètode posat a prova i dels valors reals que tenia el senyal. Abans de calcular l'MSE s'arrodoneixen els valors restaurats a l'enter més proper per adequar-los a les condicions del senyal original, que al provenir d'un sensor amb una resolució de l'1 %, no té decimals.

Es calcula l'MSE mitjançant la següent expressió, on x són les dades originals, $\hat{x}$ les restaurades, i les posicions de la ràfega de dades des de la 1 a la B (nombre total de mostres de la ràfega).

$$MSE = \frac{1}{B} \sum_{i=1}^{L} \sqrt{(x_i - \hat{x}_i)^2} \tag{3.1}$$

Es realitzen 1000 iteracions per cada simulació i es fa la mitja de l'MSE obtingut en cada una d'elles com a valor per determinar l'eficàcia del mètode i poder comparar resultats. Per generar-les s'escullen aleatòriament 1000 posicions d'inici de ràfega dins de l'historial de dades amb un mínim de separació entre elles. Per cada mètode a provar, es fan servir les mateixes 1000 posicions d'inici generades, per tant les ràfegues generades i restaurades són exactament iguals per tots ells. D'aquesta manera es poden comparar tots els mètodes en les mateixes condicions.

Capítol 4

RECONSTRUCCIÓ DE DADES AMB TÈCNIQUES LINEALS

En aquest capítol primerament s'expliquen els dos mètodes de reconstrucció de dades utilitzats fins ara pels operaris d'aigües de Vic, seccions 4.1 i 4.2, depenent de les dades perdudes i el context en que es troben.

A continuació es descriu un mètode de reconstrucció de dades que s'ha desenvolupat a partir del predictor de Wiener [13–17], secció 4.3. El mètode proposat està específicament dissenyat per restaurar dades perdudes en forma de ràfega, si es disposa d'un historial de dades posteriors i anteriors a la ràfega de dades perduda.

Finalment es recullen els resultats i les conclusions del capítol a les seccions 4.4 i 4.5.

4.1. Mètode "Dada anterior"

Aquest és un dels mètodes que fan servir els operaris d'aigües de Vic. L'utilitzen tan sols quan només disposen de la dada anterior a la ràfega de dades perdudes, i no de la següent, ja que per algun motiu, en el moment de restaurar-la no hi ha més informació disponible. Es corregeixen les mostres perdudes amb la última dada rebuda, fent servir la mateixa per tota la ràfega. És a dir, consisteix en deixar l'últim valor rebut pel sensor, fins a rebre un nova mesura vàlida. Per tant per omplir una ràfega de B mostres, que comença a x_{n+1} i acaba a x_{n+B}, només cal disposar de x_n.

$$x_{n+i} = x_n \qquad i \in 1, \cdots, B \tag{4.1}$$

Tot i la seva simplesa, en el cas del sensor de nivell estudiat, per a ràfegues curtes resulta un mètode acceptable. El problema que genera és que es provoquen graons i per tant pèrdua de la continuïtat del senyal, sobretot quan les ràfegues de mostres perdudes són llargues. Es poden veure exemples d'aquest mètode a continuació, a les figures 4.1-4.4 i en comparació amb altres mètodes a les figures 4.19-4.26.

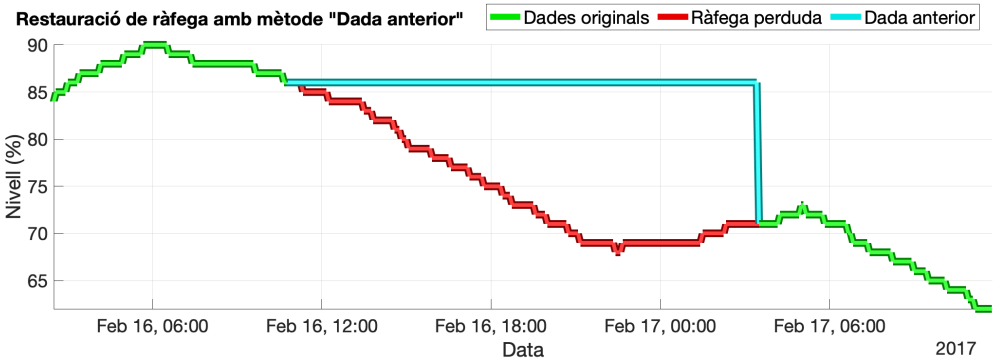

Figura 4.1: Simulació de la restauració d'una ràfega de 200 mostres amb el mètode "Dada anterior", que consisteix senzillament en mantenir el valor de la última mostra rebuda.

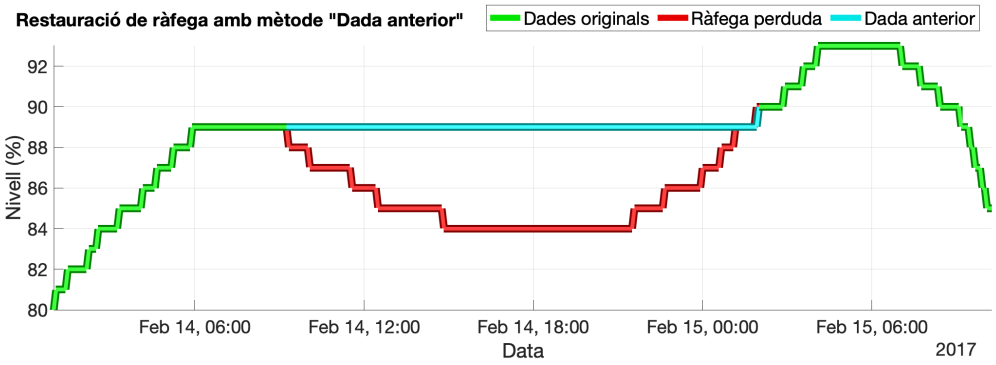

Figura 4.2: Simulació de la restauració d'una ràfega de 200 mostres amb el mètode "Dada anterior", que consisteix senzillament en mantenir el valor de la última mostra rebuda.

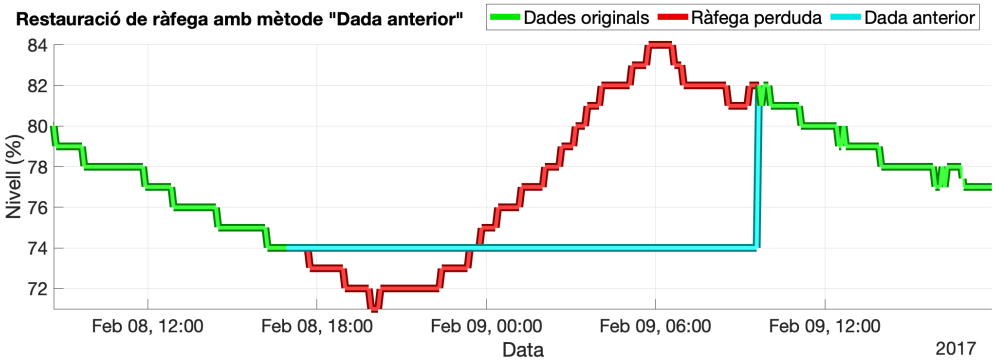

Figura 4.3: Simulació de la restauració d'una ràfega de 200 mostres amb el mètode "Dada anterior", que consisteix senzillament en mantenir el valor de la última mostra rebuda.

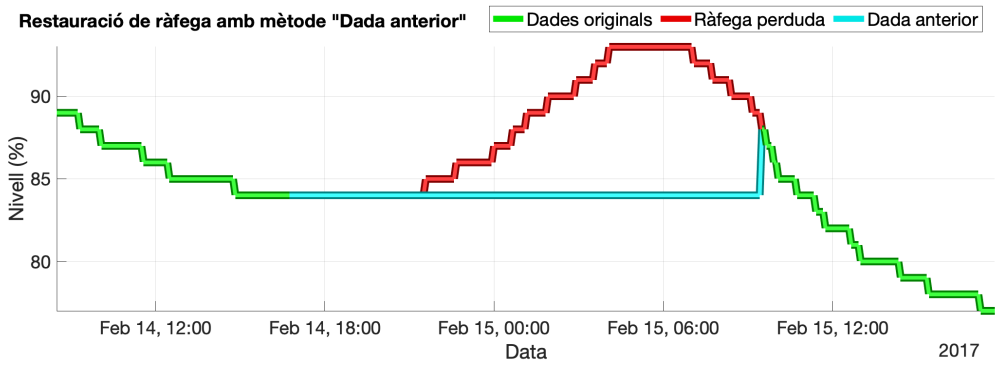

Figura 4.4: Simulació de la restauració d'una ràfega de 200 mostres amb el mètode "Dada anterior", que consisteix senzillament en mantenir el valor de la última mostra rebuda.

4.2. Mètode "Rampa"

Aquest és el mètode més elaborat que fan servir els operaris d'Aigües de Vic, tot i que en el fons és força senzill i lògic. Per a ràfegues petites dona bons resultats. Aquest mètode l'utilitzen quan disposen de mesures fiables anteriors i posteriors a la ràfega de mostres perdudes. En tal cas realitzen una aproximació força simple dibuixant una recta entre l'última mostra vàlida anterior a la ràfega de mostres perdudes, i la primera mostra vàlida posterior a la ràfega.

Per fer-ho es calcula m, que seria l'increment o decrement que cal aplicar a cada mostra perduda des de la primera mostra, x_{n+1}, fins la última mostra, x_{n+B}, on B seria la llargada total de la ràfega en número de mostres.

$$x_{n+i} = x_n + m \cdot i \quad i \in 1, \cdots, B \quad \text{on} \quad m = (x_n - x_{n+B+1}) / (B+1) \tag{4.2}$$

Tot i la seva senzillesa, aquesta forma de reconstruir la ràfega de dades perdudes evita la pèrdua de continuïtat del senyal, que en el mètode "Dada anterior" de la secció 4.1 es produïa en forma de graons. Tot i aquest avantatge, quan les ràfegues són llargues, el mètode perd efectivitat, sobretot en els casos en que es produeixen fluctuacions del senyal real durant la ràfega. Es poden veure exemples d'aquest mètode a continuació, a les figures 4.5-4.8 i en comparació amb altres mètodes a les figures 4.19-4.26.

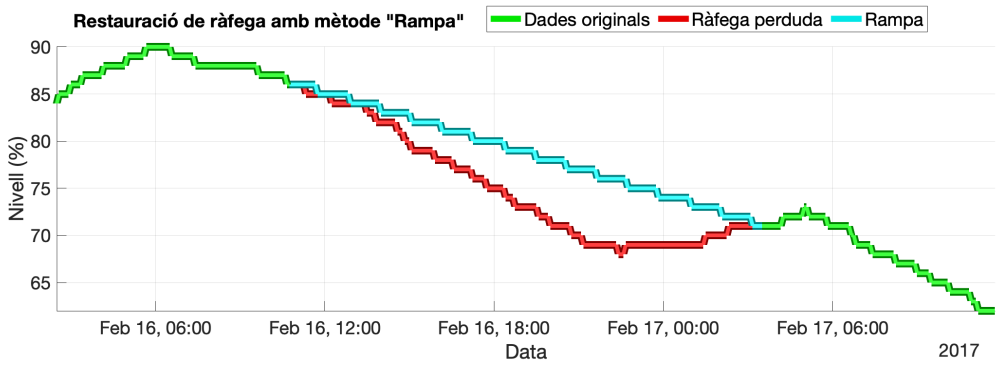

Figura 4.5: Simulació de la restauració d'una ràfega de 200 mostres amb el mètode Rampa", que consisteix en unir amb una línia recta l'últim valor rebut abans de la ràfega i el primer rebut després de la ràfega.

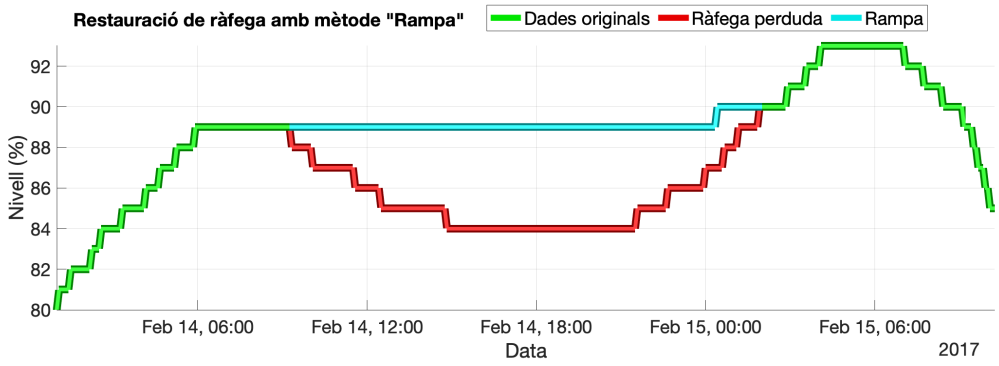

Figura 4.6: Simulació de la restauració d'una ràfega de 200 mostres amb el mètode Rampa", que consisteix en unir amb una línia recta l'últim valor rebut abans de la ràfega i el primer rebut després de la ràfega.

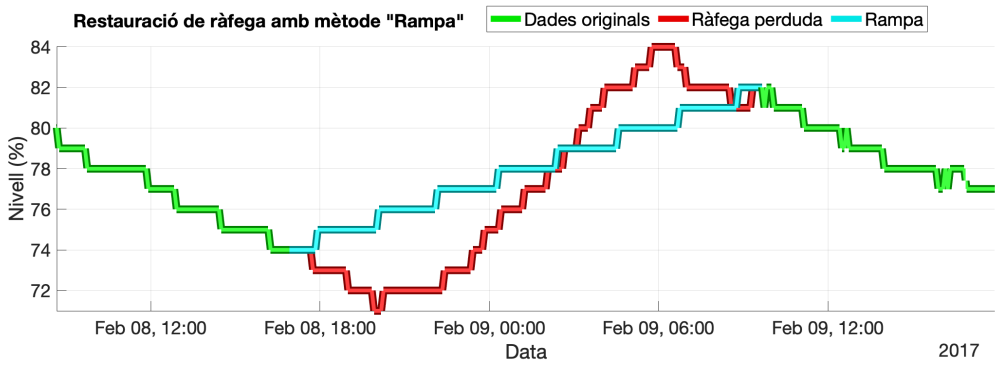

Figura 4.7: Simulació de la restauració d'una ràfega de 200 mostres amb el mètode Rampa", que consisteix en unir amb una línia recta l'últim valor rebut abans de la ràfega i el primer rebut després de la ràfega.

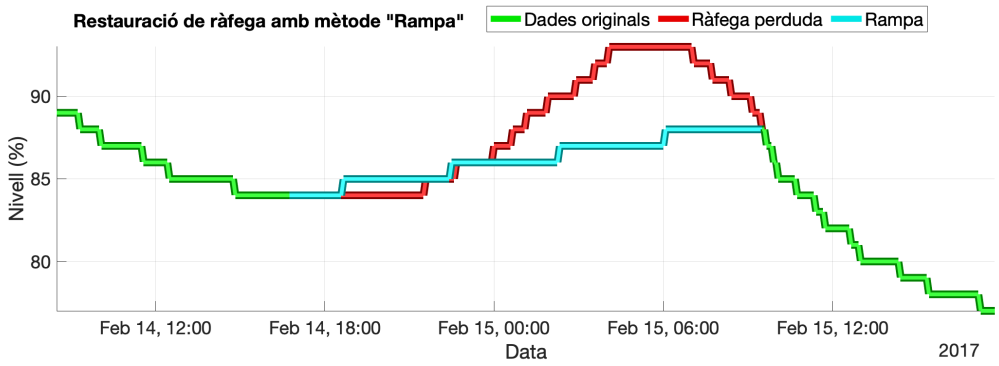

Figura 4.8: Simulació de la restauració d'una ràfega de 200 mostres amb el mètode Rampa", que consisteix en unir amb una línia recta l'últim valor rebut abans de la ràfega i el primer rebut després de la ràfega.

4.3. Mètode "FIR"

Per tal de recuperar les mostres perdudes, es fa servir el predictor de Wiener explicat a l'apartat *Predictor de Wiener* que permet predir una mostra a partir d'un conjunt de mostres anteriors. Es fa de forma, que al predir la primera mostra de la ràfega, s'utilitza aquesta mateixa predicció per predir la següent mostra de la ràfega. Repetint el procés successivament fins a recuperar tota la ràfega de dades perdudes. Com que es disposa d'un historial de mostres, anteriors i posteriors a la ràfega de dades perdudes, a l'apartat *Combinació de FIR endavant i enrere* es combinen de forma ponderada dos versions del predictor de Wiener, una que treballa amb les mostres anteriors a la ràfega, i una altra que ho fa de forma recíproca amb les dades posteriors a la ràfega, per obtenir una reconstrucció de la ràfega que no perd la continuïtat del senyal. A l'apartat *Configuració de L i M* es mostra com configurar el mètode i es poden veure exemples d'aquest mètode a les figures 4.10-4.26.

4.3.1. Predictor de Wiener

El predictor de Wiener es dissenya mitjançant criteris estadístics. Es calculen els coeficients a_i òptims d'un filtre FIR (Finite Impulse Response o Resposta Impulsional Finita) a partir de minimitzar la funció de cost $E[|e_n|^2]$, on e_n és l'error estimat de predicció, $E[\cdot]$ és l'operador esperança i $|\cdot|$ la norma euclidiana.

L'error estimat de predicció és la diferència entre la mostra original x_n i la

mostra predita $\hat{x}_n$, que s'obté aplicant el filtre FIR (d'ordre W), a les W dades anteriors a la mostra x_n. Com es mostra en el diagrama de la figura 4.9.

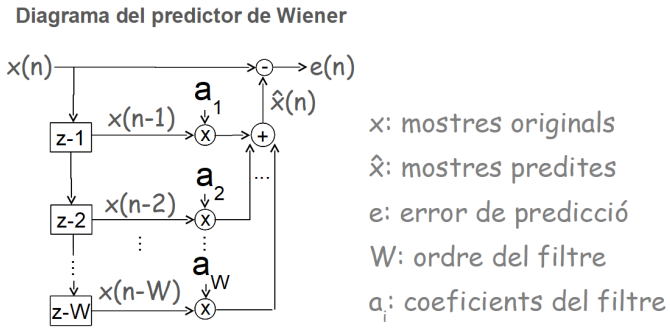

Figura 4.9: Diagrama del predictor de Wiener.

Com es mostra en el diagrama de forma esquemàtica, en termes vectorials es pot definir l'error com $e_n = x_n - \hat{x}_n = x_n - \boldsymbol{a_i}^T \boldsymbol{x_{n-1}}$ on el vector $\boldsymbol{a_i}^T = [a_1 \dots a_L]$ conté els L coeficients del filtre i el vector $\boldsymbol{x_{n-1}}^T = [x_{n-1} \dots x_{n-L}]$ conté les mostres usades per fer les aproximacions, les L anteriors a la que es vol predir. Seguint la lògica de l'indexat $x_n = [x_n \dots x_{n-L+1}]$. L'ordre del filtre L, que indica el nombre de coeficients usats, determina la mida dels vectors. De fet també determina el número de mostres anteriors a la que es vol predir que seran necessàries per fer la predicció.

Per minimitzar l'MSE es formula la funció de cost següent:

$$E[|e_n|^2] = E[e_n e_n^T] = E[(x_n - \boldsymbol{a_i}^T \boldsymbol{x_{n-1}})(x_n - \boldsymbol{a_i}^T \boldsymbol{x_{n-1}})^T] \tag{4.3}$$

$$E[|e_n|^2] = E[x_n x_n^T - x_n \boldsymbol{x_{n-1}}^T \boldsymbol{a_i} - \boldsymbol{a_i}^T \boldsymbol{x_{n-1}}^T x_n + \boldsymbol{a_i}^T \boldsymbol{x_{n-1}} \boldsymbol{x_{n-1}}^T \boldsymbol{a_i}] \tag{4.4}$$

$$E[|e_n|^2] = E[x_n x_n] - 2E[x_n \boldsymbol{x_{n-1}}^T] \boldsymbol{a_i} + \boldsymbol{a_i}^T E[\boldsymbol{x_{n-1}} \boldsymbol{x_{n-1}}^T] \boldsymbol{a_i} \tag{4.5}$$

Seguint el criteri de buscar els coeficients que proporcionen el mínim MSE cal resoldre l'equació següent:

$$\frac{\partial E[|e_n|^2]}{\partial \boldsymbol{a_i}} = 0, \tag{4.6}$$

Obtenint:

$$\frac{\partial E[|e_n|^2]}{\partial \boldsymbol{a_i}} = -E[x_n \boldsymbol{x_{n-1}}^T] + E[\boldsymbol{x_{n-1}} \boldsymbol{x_{n-1}}^T] \boldsymbol{a_i} \tag{4.7}$$

Per tant els coeficients del filtre es calculen de la següent manera:

$$\boldsymbol{a_i} = E[\boldsymbol{x_{n-1}} \boldsymbol{x_{n-1}}^T]^{-1} E[x_n \boldsymbol{x_{n-1}}^T] \tag{4.8}$$

El factor $E[\boldsymbol{x_{n-1}}\boldsymbol{x_{n-1}}^T]^{-1}$ es equivalent a $\boldsymbol{R_{xx}}^{-1}$, que és la inversa de la matriu d'auto-correlació de $\boldsymbol{x_{n-1}}$ respecte $\boldsymbol{x_{n-1}}$. Per valors reals $\boldsymbol{R_{xx}}^{-1}$ es una matriu Toeplitz simètrica:

$$\boldsymbol{R_{xx}} = \begin{bmatrix} r_0 & \dots & r_{L-1} \\ \vdots & \ddots & \vdots \\ r_{L-1} & \dots & r_0 \end{bmatrix} \tag{4.9}$$

L'altre factor pel càlcul dels coeficients $E[x_n\boldsymbol{x_{n-1}}^T]$ es pot escriure com $\boldsymbol{r}_{x\boldsymbol{x_{n-1}}}$ que és el vector de correlació de x_n respecte de x_{n-1}:

$$\boldsymbol{r}_{x\boldsymbol{x_{n-1}}} = [r_1 \dots r_L] \tag{4.10}$$

Finalment l'expressió per calcular els coeficients del filtre és:

$$\boldsymbol{a_i} = \boldsymbol{R_{xx}}^{-1}\boldsymbol{r}_{x\boldsymbol{x_{n-1}}}. \tag{4.11}$$

Pel que fa al disseny del predictor de Wiener es pot dir que hi ha dos punts clau. El primer fa referència a la mida del filtre, el nombre de coeficients L [30]. El segon fa referència al nombre de mostres, utilitzades per estimar els coeficients d'auto-correlació r_i [31]. Que queda definit en la finestra de mida M segons l'expressió :

$$r_k = \frac{\sum_{i=1}^{M}(x_i - \bar{x})(x_{i-k} - \bar{x})}{\sum_{i=1}^{M}(x_i - \bar{x})^2}, \tag{4.12}$$

On:

$$\bar{x} = \frac{1}{M}\sum_{i=1}^{M} x_i \tag{4.13}$$

4.3.2. Combinació de FIR endavant i enrere

Aquest mètode es basa en utilitzar la combinació de dos versions del predictor de Wiener (que fa servir un filtre FIR).

La primera versió, que seria la manera clàssica de fer prediccions amb el filtre de Wiener, comença per estimar la primera mostra de la ràfega de dades perdudes. Es realitza l'estimació d'aquetsa primera mostra $\hat{x}(n)$ mitjançant l'expressió vectorial $\boldsymbol{a_i}^T\boldsymbol{x_{n-1}}$. Tot seguit, aquesta estimació $\hat{x}(n)$ s'introdueix en el vector $\boldsymbol{x_{n-1}}$, amb la idea d'actualitzar el vector i poder estimar la mostra següent. Mitjançant la repetició d'aquest procés s'aconsegueix omplir tota la

ràfega de dades perdudes. Amb aquest sistema, encara que les dades estiguin molt correlades, a mesura que es van estimant mostres l'error d'estimació va creixent. Degut a això al predir tota la ràfega, és molt difícil que l'ultima mostra estimada correspongui amb la primera mostra disponible després de la ràfega, amb lo qual probablement es perd la continuïtat del senyal. Anomenem aquesta metodologia, estimació "FIR endavant".

Tenint en compte el fenomen descrit, es desenvolupa una nova versió, l'estimació "FIR enrere" que consisteix en repetir la mateixa tècnica però utilitzant les mostres de després de la ràfega. De forma que es comença a estimar la ràfega pel final i s'acaba al principi, progressant enrere en el temps. Amb aquest canvi apareix un efecte recíproc a l'anterior mètode, les estimacions del final de la ràfega, tenen un error més petit que les del principi, de forma que probablement es perd la continuïtat al principi de la ràfega, però mai al final.

Els dos mètodes proposats, perden la continuïtat del senyal en un dels extrems, però capturen força bé les fluctuacions del senyal. Per aprofitar els dos mètodes i no perdre la continuïtat del senyal es combinen de la següent manera:

$$\text{for} \quad i = 1 \dots N; \qquad \hat{x}_i = \begin{cases} \dfrac{\hat{f}_i + \hat{b}_i}{2} & \text{if } N = 1 \\ \dfrac{(N-i)\,\hat{f}_i + (i-1)\,\hat{b}_i}{N-1} & \text{if } N > 1 \end{cases} \tag{4.14}$$

On i és l'index corresponent a les N mostres perdudes del principi al final segons l'ordre cronològic. Amb aquesta expressió es dona més pes a la millor de les estimacions en cada extrem de la ràfega, evitant la pèrdua de la continuïtat.

Les figures 4.10-4.13 mostren exemples de reconstruccions de ràfegues fetes aplicant aquesta metodologia en les simulacions que s'han dut a terme. A les figures 4.14-4.17 es mostren exemples del mateix mètode aplicat a casos reals de ràfegues de dades perdudes. Es pot observar com les estimacions "FIR endavant" i "FIR enrere" segueixen la forma del senyal, tot i que perden la continuïtat del mateix. Però amb la combinació de les dues versions del mètode "FIR"definitiu això s'evita, i s'obté una estimació molt més consistent.

Finalment també es mostren alguns exemples d'aquest mètode en simulacions i en casos reals i en comparació amb els altres mètodes, els que fan servir els operaris, a les figures 4.19-4.26.

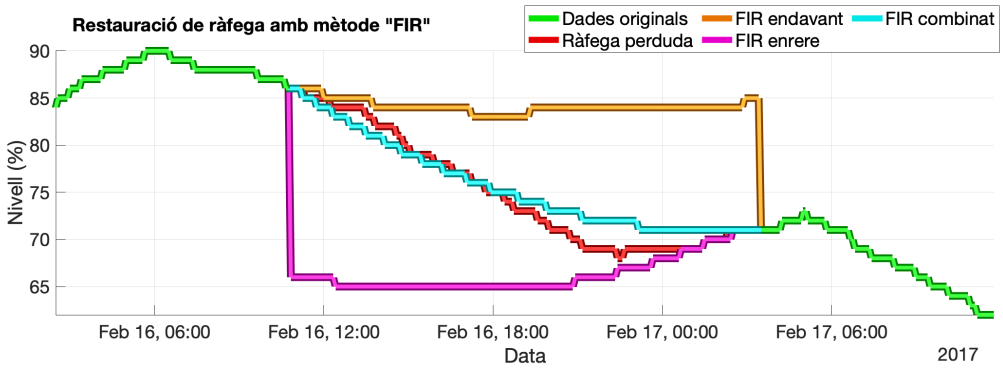

Figura 4.10: Simulació de la restauració d'una ràfega de 200 mostres amb el mètode "FIR". "FIR endavant" usa dades anteriors a la ràfega per estimar-la des de l'inici fins al final i "FIR enrere" usa les posteriors per fer-ho recíprocament.

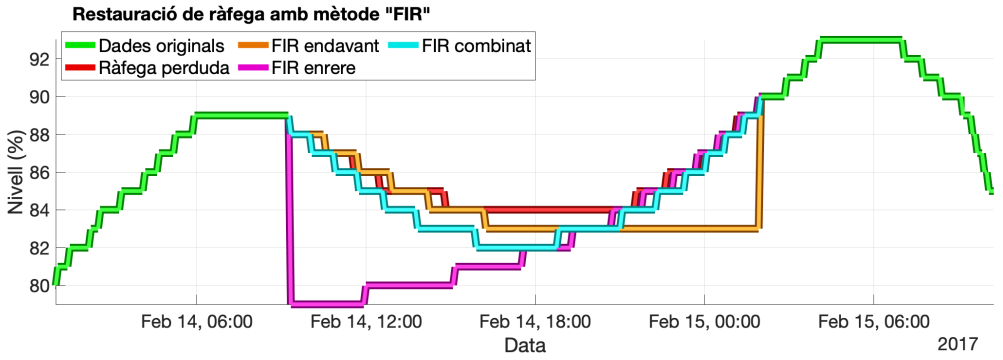

Figura 4.11: Simulació de la restauració d'una ràfega de 200 mostres amb el mètode "FIR". "FIR endavant" usa dades anteriors a la ràfega per estimar-la des de l'inici fins al final i "FIR enrere" usa les posteriors per fer-ho recíprocament.

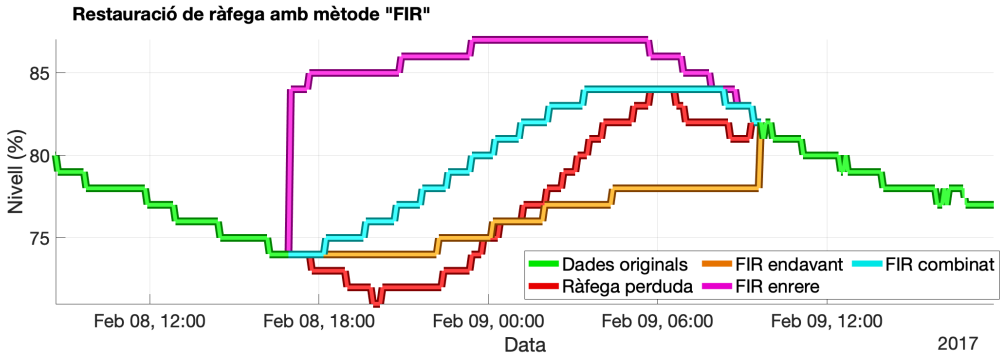

Figura 4.12: Simulació de la restauració d'una ràfega de 200 mostres amb el mètode "FIR". "FIR endavant" usa dades anteriors a la ràfega per estimar-la des de l'inici fins al final i "FIR enrere" usa les posteriors per fer-ho recíprocament.

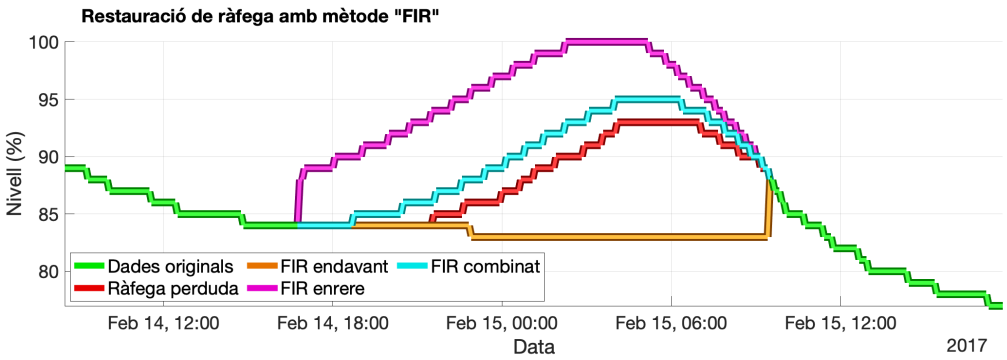

Figura 4.13: Simulació de la restauració d'una ràfega de 200 mostres amb el mètode "FIR". "FIR endavant" usa dades anteriors a la ràfega per estimar-la des de l'inici fins al final i "FIR enrere" usa les posteriors per fer-ho recíprocament.

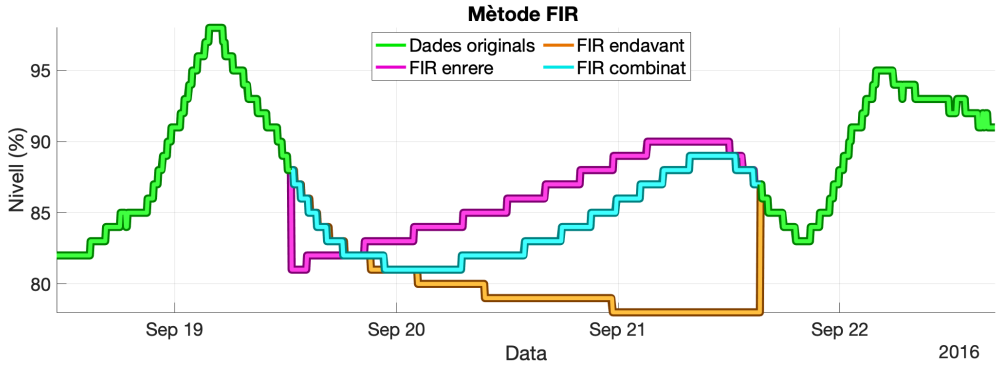

Figura 4.14: Cas real de la restauració d'una ràfega amb el mètode "FIR". "FIR endavant" usa dades anteriors a la ràfega per estimar-la des de l'inici fins al final i "FIR enrere" usa les posteriors per fer-ho recíprocament.

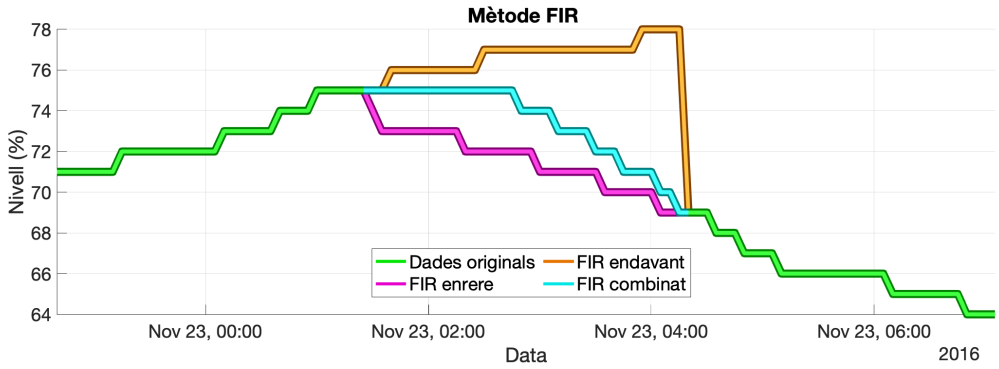

Figura 4.15: Cas real de la restauració d'una ràfega amb el mètode "FIR". "FIR endavant" usa dades anteriors a la ràfega per estimar-la des de l'inici fins al final i "FIR enrere" usa les posteriors per fer-ho recíprocament.

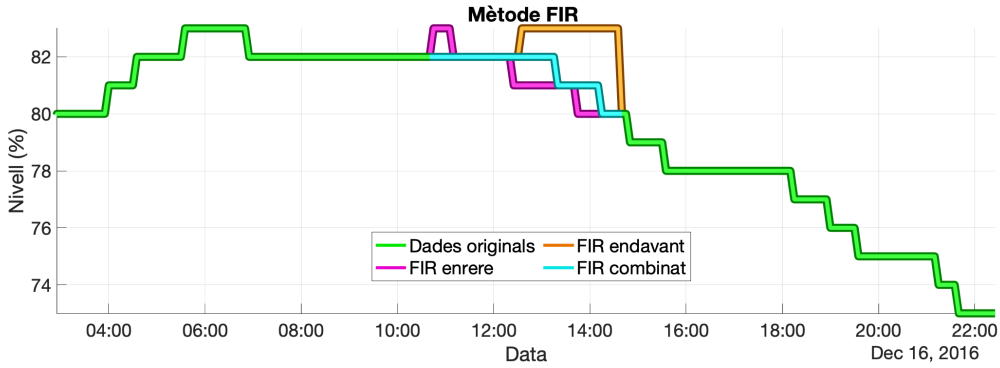

Figura 4.16: Cas real de la restauració d'una ràfega amb el mètode "FIR". "FIR endavant" usa dades anteriors a la ràfega per estimar-la des de l'inici fins al final i "FIR enrere" usa les posteriors per fer-ho recíprocament.

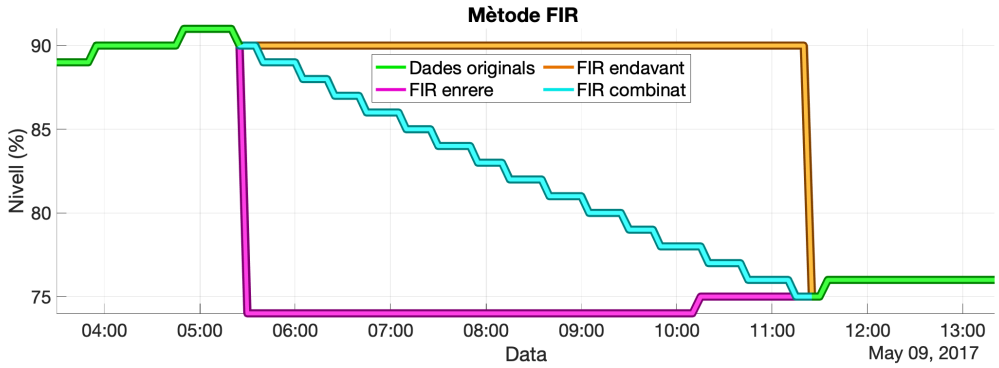

Figura 4.17: Cas real de la restauració d'una ràfega amb el mètode "FIR". "FIR endavant" usa dades anteriors a la ràfega per estimar-la des de l'inici fins al final i "FIR enrere" usa les posteriors per fer-ho recíprocament.

4.3.3. Configuració de L i M

Els dos mètodes utilitzats pels operaris, no tenen cap configuració ja que són molt senzills i pràcticament no depenen de cap paràmetre, només de l'historial de mostres registrades de la senyal a restaurar. El predictor de Wiener, en canvi, que és utilitzat en el mètode proposat, sí que depèn de vairs paràmetres, a part de les mostres de l'historial, i per tant es pot configurar. En concret es poden configurar L que seria la mida del filtre i M que seria el número de mostres utilitzades per calcular la correlació del senyal. Es realitza un experiment amb simulacions d'aquest mètode per a diferents combinacions de la configuració d'aquests dos paràmetres, amb l'objectiu de determinar estadísticament quina és la millor configuració, és a dir la millor configuració d'aquests paràmetres en el nostre cas. Els resultats es mostren a la figura 4.18. S'aplica el criteri

d'escollir els mínims valors per L i M amb un MSE com a molt un 1 % més gran que el mínim assolit, resultant com a combinació seleccionada $L = 65$ i $M = 2000$. S'han inclòs altres combinacions en algunes de les simulacions per veure'n els efectes, amb un criteri semblant però elevant el marge d'error a 5 %, 10 % i 15 %, obtenint (20/1500), (4/150) i (20/250) com a (L/M) respectivament.

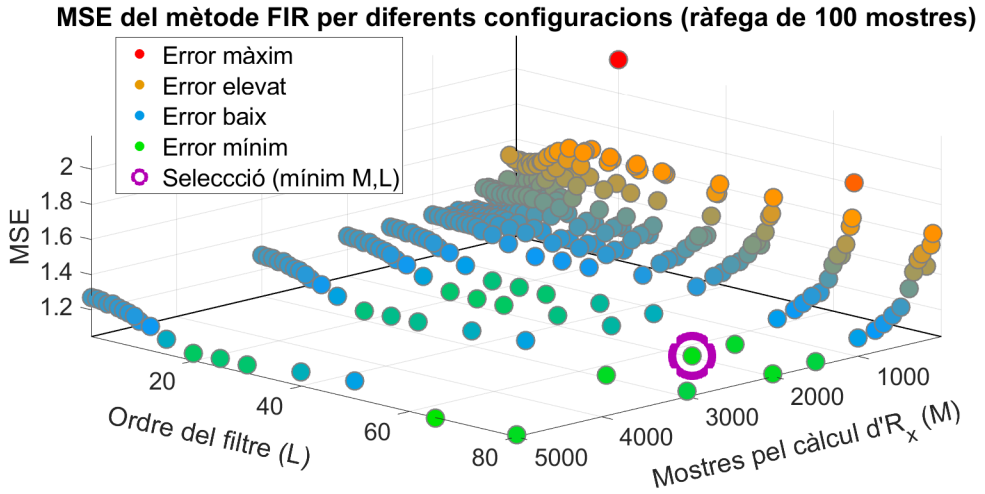

Figura 4.18: MSE del mètode "FIR" segons la configuració de l'ordre del filtre L i el nombre de mostres usades en el càlcul de la correlació del senyal M.

4.4. Resultats

Després d'analitzar els mètodes que fan servir els operaris d'Aigües de Vic per restaurar dades perdudes s'observa que, aplicant-los al cas del sensor de nivell del dipòsit principal de Vic, en casos concrets on es perden un llarg conjunt de mostres de forma consecutiva, perden molta eficàcia. Es desenvolupa la metodologia de la secció 4.3 per mirar d'oferir una eina que ajudi a restaurar les dades en aquests casos. A les figures 4.14-4.17 es mostren alguns exemples del funcionament del mètode desenvolupat a partir de dos versions del predictor de Wiener. S'experimenta amb la configuració del mètode proposat a l'apartat *Configuració de L i M* de la secció 4.3 per determinar els valors òptims de configuració que permetin obtenir els millors resultats. En l'experiment realitzat resulten ser $L = 65$ per l'ordre del filtre i pel nombre de mostres utilitzades en el càlcul de la correlació $M = 2000$. Fent que el filtre òptim pel sensor de nivell en les condicions descrites sigui el que anomenarem FIR(65/2000). A les figures 4.23-4.26 es mostren alguns exemples amb reconstruccions de ràfegues de dades perdudes realment.

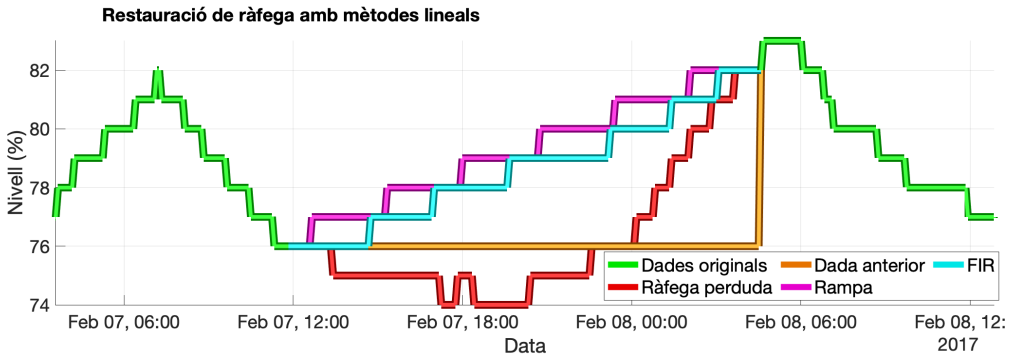

Figura 4.19: Simulació de la restauració d'una ràfega de 200 mostres amb els mètodes lineals d'aquest capítol.

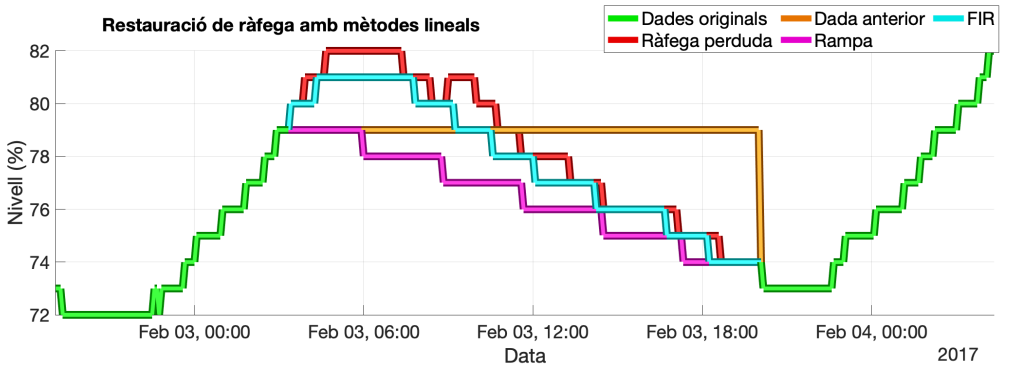

Figura 4.20: Simulació de la restauració d'una ràfega de 200 mostres amb els mètodes lineals d'aquest capítol.

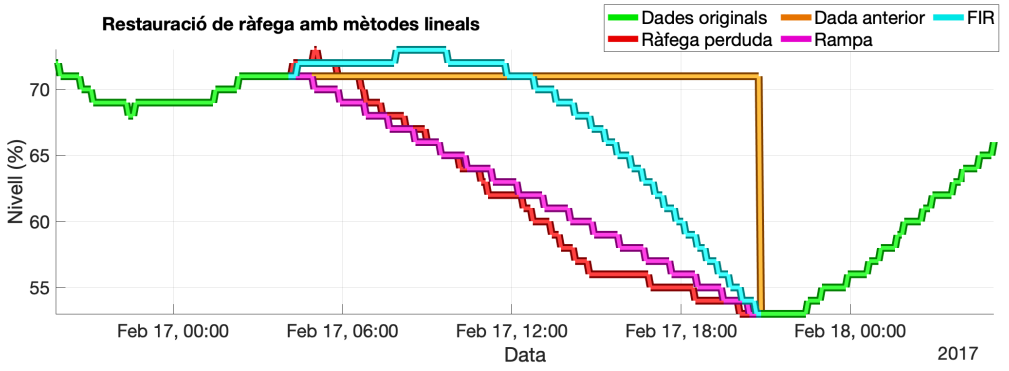

Figura 4.21: Simulació de la restauració d'una ràfega de 200 mostres amb els mètodes lineals d'aquest capítol.

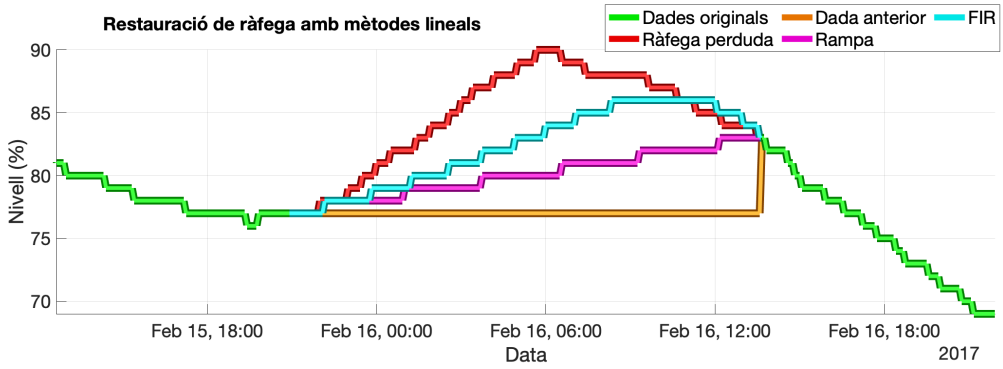

Figura 4.22: Simulació de la restauració d'una ràfega de 200 mostres amb els mètodes lineals d'aquest capítol.

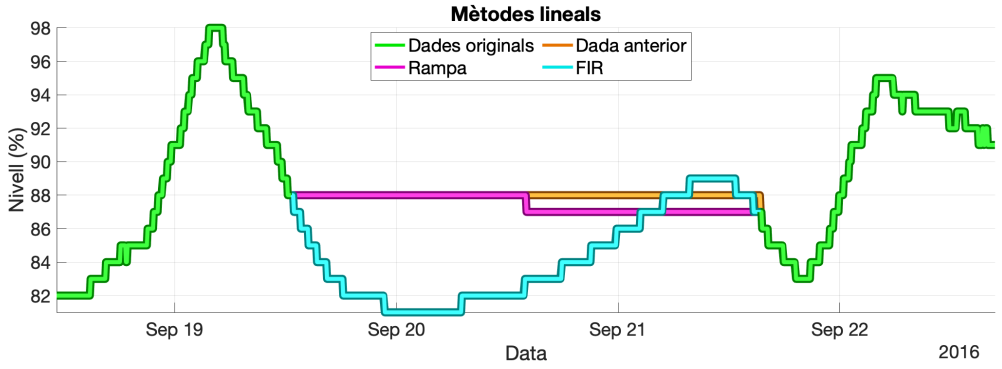

Figura 4.23: Cas real de la restauració d'una ràfega amb els mètodes lineals d'aquest capítol.

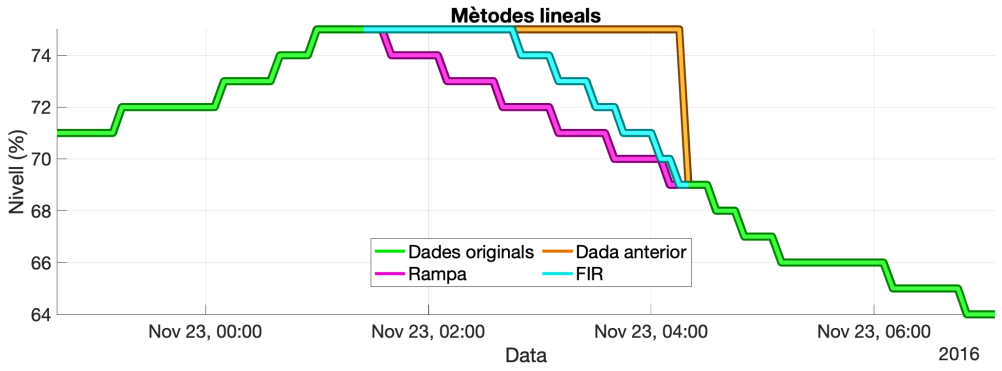

Figura 4.24: Cas real de la restauració d'una ràfega amb els mètodes lineals d'aquest capítol.

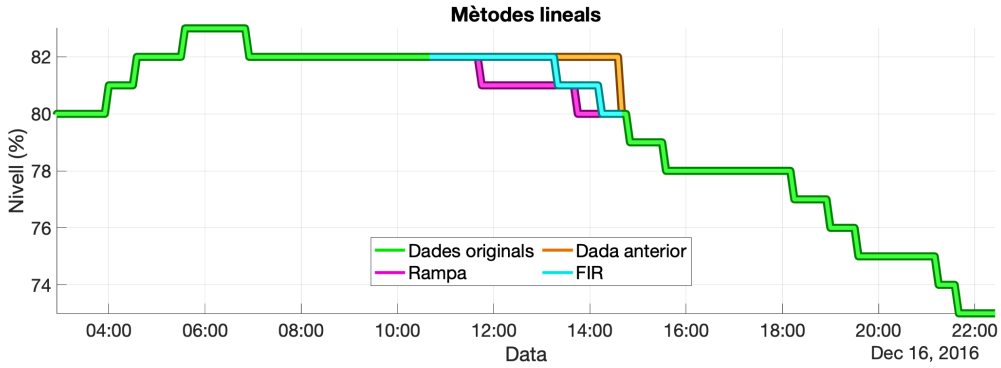

Figura 4.25: Cas real de la restauració d'una ràfega amb els mètodes lineals d'aquest capítol.

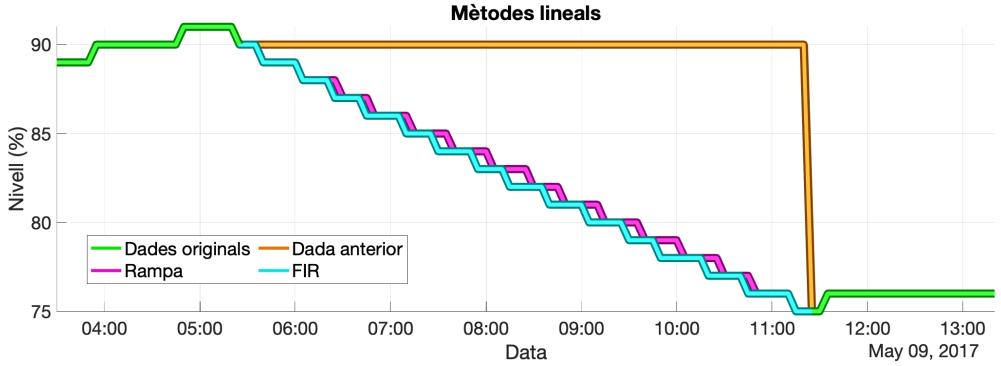

Figura 4.26: Cas real de la restauració d'una ràfega amb els mètodes lineals d'aquest capítol.

4.5. Conclusions

Hi ha diferents tècniques de predicció [4], i de reconstrucció de dades [3, 32, 33]. A part del filtre FIR del predictor de Wiener, un altre filtre lineal molt utilitzat és l'anomenat filtre de Kalman [34, 35]. En general hi ha molts exemples d'aplicacions que utilitzen tècniques lineals per la reconstrucció de dades, com són [36, 37] en l'àmbit de la qualitat de l'aire, o [38] en el de l'activitat sísmica.

Les dades més difícils de reconstruir són les que es produeixen en ràfegues llargues, normalment degudes a fallades de comunicació o del sensor. Quan una fallada es produeix, els operaris o el servei tècnic corregeixen el problema al més ràpid possible, fent que la pèrdua de dades no solgui ser de més d'un dia, però si de vàries hores.

La metodologia proposada a la secció 4.3 ofereix millores en els resultats de

les reconstruccions quan les ràfegues són més llargues. A més, igual que el cas del mètode de la secció 4.2 usat pels operaris, el mètode proposat anomenat "FIR" no perd la continuïtat del senyal. Pel que a fa a les ràfegues més llargues, el mètode "FIR" manté la coherència del senyal en tots els casos, encara que no realitzi una aproximació exacte, com es pot apreciar a la figura 4.23.

Un detall en contra d'aquesta metodologia, és que el temps d'excussió és força elevat. Per fer-se una idea, amb un Intel(R) Core(TM) i5-6200U de 2.3 GHz i 8GB de RAM amb Windows 7 Professional i utilitzant el Matlab 2018, el mètodes senzills dels operaris que pràcticament no requereixen càlculs, "Dada Anterior" i "Rampa", tarden només 0,2 i 4 ms respectivament. El mètode lineal "FIR" proposat, en canvi, demana de 2,5 a 10 segons, principalment degut al càlcul de la correlació i de la matriu inversa.

Capítol 5

MÈTODE DE RECONSTRUCCIÓ DE DADES AMB TENSORS

En aquest capítol es presenta el primer mètode de recuperació de dades proposat que fa ús de l'àlgebra dels tensors. La primeres seccions del capítols es dediquen a explicar els conceptes bàsics i el mètode, mentre que les últimes es dediquen a la presentació dels resultats i conclusions.

A la secció 5.1 s'introdueix el concepte de tensor, que es farà servir per desenvolupar un mètode de reconstrucció de dades. Primer s'explica la idea fonamental del que són els tensors a l'apartat *Conceptes bàsics*. Després, a l'apartat *Models Tucker i CANDECOMP/PARAFAC*, es detallen els models que es faran servir per la descomposició del tensor en el mètode de reconstrucció de dades proposat. També es mostra el cas particular de l'algoritme CP Weighted Optimization (CP-Wopt), ja que els seus autors han lliurat el codi per a ser executat en Matlab, apartat *Algoritme CP-Wopt*.

A la secció 5.2, s'explica la metodologia proposada per a la restauració de dades utilitzant tensors. Es detalla com s'organitzen les dades en el tensor que es fa servir pels experiments a l'apartat *Introducció de les dades al tensor* i, a l'apartat *Procediment*, es descriuen les operacions que es segueixen per aprofitar l'àlgebra tensorial i estimar les dades perdudes. També s'explica l'algoritme que s'aplica per no perdre la continuïtat del senyal a l'apartat *Correcció de la continuïtat*.

Un cop descrita la metodologia proposada, a la secció 5.3, es mostra com configurar els paràmetres per optimitzar els resultats. A l'apartat *Nucli de la descomposició* es pot veure com configurar el nucli de la descomposició tensorial. Als apartats *Mida del tensor* i *Mida de la ràfega* s'analitza l'efecte de les mides del tensor i de la ràfega respectivament.

Finalment, com s'ha comentat, es recullen els resultats i les conclusions del capítol a les seccions 5.4 i 5.5.

5.1. Tensors

5.1.1. Conceptes bàsics

Un tensor es pot entendre com un contenidor que permet organitzar les dades en N-dimensions [9, 18, 39, 40]. El nombre de dimensions, N, és el que s'anomena l'ordre del tensor. El cas d'un tensor de números reals d'N-dimensions s'escriu com $\chi \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. Segons això un vector $\mathbf{x}$, $N \times 1$, es podria considerar un tensor d'ordre 1 i, una matriu $\mathbf{X}$, $N \times M$, un tensor d'ordre 2.

Un "sub-tensor" és una part del tensor original, definit per uns índex determinats. Per exemple, si es fixen tots els índex del tensor excepte un, s'obté un vector anomenat "fiber" (fibra). En el cas concret d'un tensor de tres dimensions, $\chi_{:,j,k}$, $\chi_{i,:,k}$, i $\chi_{i,j,:}$ en serien fibres. Si es fixen tots els índex excepte dos, s'obté una matriu anomenada "slice" (llesca o tall). En el mateix cas les llesques serien $\chi_{:,:,k}$, $\chi_{:,j,:}$ i $\chi_{i,:,:}$.

El procés de transformar tensors en matrius o vectors juga un paper important, sobretot al moment de definir les operacions algebraiques entre ells.

Sovint s'usa la notació $\mathbf{X}_{(n)}$ per representar la matriu *mode-n* de $\chi \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, operació amb la que es transforma χ en la matriu $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times I_1 I_2 \cdots I_{n-1} I_{n+1} \cdots I_N}$. De la mateixa manera es fa servir $vec(\chi)$ que transforma $\chi \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ en el vector $\boldsymbol{x} \in \mathbb{R}^{I_1 I_2 \cdots I_N}$.

L'àlgebra tensorial té algunes similituds, però també grans diferencies amb l'àlgebra matricial. Particularment és important definir el producte entre matrius i tensors per tal d'entendre la descomposició del tensor que es fa servir.

El producte mode-n d'un tensor $\chi \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n \times \cdots \times I_N}$ per una matriu $\mathbf{A} \in \mathbb{R}^{J_n \times I_n}$ s'escriu com $\zeta = \chi \times_n \mathbf{A}$, amb $\zeta \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times J_n \times \cdots \times I_N}$ com a tensor resultat.

El producte mode-n $\zeta = \chi \times_n \mathbf{A}$ es pot representar de forma matricial com a $\mathbf{Z}_{(n)} = \mathbf{A}\mathbf{X}_{(n)}$, on $\mathbf{Z}_{(n)}$ i $\mathbf{X}_{(n)}$ són les matrius mode-n dels tensors ζ i χ, respectivament.

Totes aquestes operacions i algunes més s'expliquen detalladament i amb exemples gràfics a [8, 9, 18, 41–45].

5.1.2. Models Tucker i CANDECOMP/PARAFAC

En molts casos, especialment en problemes on hi participen múltiples variables, l'estructura de dades manté patrons (regularitats) entre dimensions. En el tractament d'aquestes dades, quan s'utilitzen aproximacions matricials clàssiques, part de l'estructura multi-dimensional es perd, però les tècniques de descomposició de tensors permeten explotar l'estructura multi-dimensional. Tot i que hi ha un bon ventall de descomposicions tensorials existents, n'hi ha dues de molt utilitzades, que són les que es faran servir en aquesta tesi: la descomposició de Tucker [10] i la descomposició canònica també anomenada factorització paral·lela coneguda com CANDECOMP/PARRAFAC (CP) [11, 12]. Aquesta segona factorització va ser trobada per dos autors de forma independent, de forma que ha conservat els dos noms. De fet la descomposició CP és un cas particular de la Tucker, que es dóna quan el tensor de la descomposició és diagonal. Per fer l'explicació més senzilla es consideraran descomposicions d'ordre 3, que a més són les que s'utilitzen en la tesi.

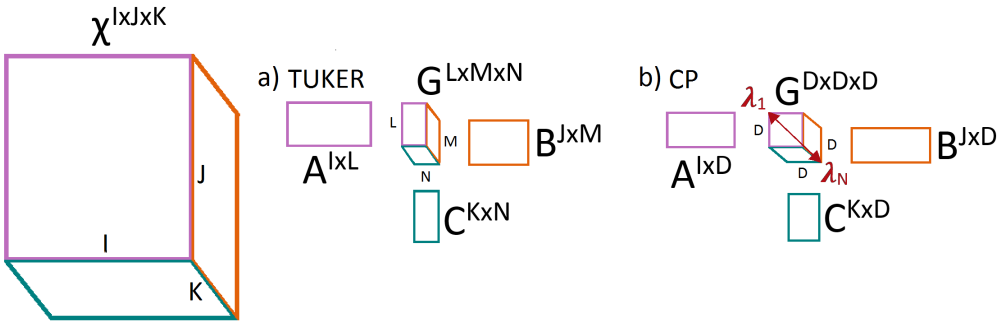

Figura 5.1: Diagrama dels models de Tensors. a) model Tucker. b) model CANDECOMP/PARAFAC (CP).

La descomposició amb el model Tucker representada a la figura 5.1 aproxima el tensor d'ordre 3 $\chi^{I\times J\times K}$ en un producte d'un nucli $G^{L\times M\times N}$ per les matrius $\boldsymbol{A}^{I\times L}$, $\boldsymbol{B}^{J\times M}$ i $\boldsymbol{C}^{K\times N}$, cada una en la seva dimensió respectiva. El tensor del nucli, $G^{L\times M\times N}$, té mida inferior a la de l'original $\chi^{I\times J\times K}$ i els seus elements indiquen la interacció els modes de cada dimensió.

Aclarim que els modes de cada dimensió són els vectors columna de les matrius $\boldsymbol{A}^{I\times L}$, $\boldsymbol{B}^{J\times M}$ i $\boldsymbol{C}^{K\times N}$. És comú fer servir la notació Tucker(L, M, N) per indicar el nombre de modes de cada dimensió. Fent servir el producte entre tensor i matriu, la descomposició resulta:

$$\chi^{I\times J\times K} \approx G^{L\times M\times N} \times_1 \boldsymbol{A}^{I\times L} \times_2 \boldsymbol{B}^{J\times M} \times_3 \boldsymbol{C}^{K\times N} \tag{5.1}$$

La figura 5.1(a) mostra gràficament el model de la descomposició Tucker(L, M, N), representant el tensor i el producte de matrius per les tres dimensions.

Com s'ha comentat la descomposició CP és un cas particular de la Tucker on les dimensions del nucli són la mateixa, L = M = N (= D). Les interaccions amb el model CP es produeixen només entre vectors amb el mateix index, cosa que implica que el nucli tensorial $D^{D\times D\times D}$ ha de ser diagonal o sigui que a la diagonal hi ha els elements que no són 0 ($d_{l,m,n} \neq 0$ només si l = m = n). El model CP en termes de producte entre tensor i matriu es pot escriure com:

$$\chi^{I\times J\times K} \approx D^{D\times D\times D} \times_1 \boldsymbol{A}^{I\times D} \times_2 \boldsymbol{B}^{J\times D} \times_3 \boldsymbol{C}^{K\times D}. \tag{5.2}$$

La descomposició CP també es pot escriure en termes de *producte exterior*, $\circ$, típicament anomenat *outer product*, entre els vectors $\mathbf{a}_i$, $\mathbf{b}_i$ i $\mathbf{c}_i$ tal com:

$$\chi^{I\times J\times K} \approx \sum_{i=1}^{D} \lambda_i \mathbf{a}_i \circ \mathbf{b}_i \circ \mathbf{c}_i \tag{5.3}$$

On els vectors $\mathbf{a}_i$, $\mathbf{b}_i$ i $\mathbf{c}_i$ estan relacionats amb les matrius de l'equació (5.2) segons: $\boldsymbol{A}^{I\times D} = [\mathbf{a}_1 \cdots \mathbf{a}_D]$, $\boldsymbol{B}^{J\times D} = [\mathbf{b}_1 \cdots \mathbf{b}_D]$ i $\boldsymbol{C}^{K\times D} = [\mathbf{c}_1 \cdots \mathbf{c}_D]$.

Atesa que aquesta descomposició només depèn del paràmetre D la referenciem amb la notació CP(D). La figura 5.1(b) mostra gràficament el model de la descomposició CP(D) d'acord a les dues formulacions explicades.

5.1.3. Algoritme CP-Wopt

A [46] es presenta una modificació de la descomposició CP que es pot utilitzar en tensors amb dades perdudes indicant quines són, de forma que el model ignora aquests valors. L'algoritme utilitza una aproximació optimitzada de primer ordre per resoldre el problema ponderat de mínims quadrats, coneguda com a CP-Wopt (CP Weighted OPTimization). És un dels mètodes de restauració de dades més usats ja que s'ha comprovat que pot ser extremadament bo recuperant dades perdudes, en comparació a mètodes de dos dimensions [46]. És important notar que l'eficàcia de l'algorisme es manifesta quan les dades perdudes estan repartides de forma escampada dins el tensor, és a dir, més o meny repartides. L'algoritme CP-Wopt s'inclou a la llibreria Poblano de Matlab [28], per tant es va provar fàcilment mitjançant el test descrit a la secció 3.4.

En el cas que ens ocupa, les pèrdues de les dades es produeixen a ràfegues i la seva capacitat de recuperació decau considerablement. L'algoritme retorna un

tensor amb els valors estimats i tot i que el senyal recuperat es força similar al original, i segueix força bé les seves fluctuacions, també tendeix a patir un cert desplaçament que suposa un MSE elevat, sobretot en els extrems del tensor.

Es mostren exemples de la reconstrucció del senyal de nivell amb aquest mètode fent servir un tensor $\chi^{288\times7\times3}$ a les figures 5.2, 5.3. En general es pot veure com el senyal recuperat (color cian) és una mica diferent del original (color blau fort), tot i que segueix força bé les seves fluctuacions. A la figura 5.2 es mostren dos exemples en els que la ràfega recuperada (color vermell) coincideix força bé amb les dades originals. En canvi a la figura 5.3 se'n mostren dos en els que la ràfega recuperada té un error molt gran. Si s'observen amb atenció les mostres concretes de la ràfega perduda (les vermelles) es pot apreciar que l'error és sobretot un desplaçament, o un "offset", i que la forma del senyal segueix essent semblant.

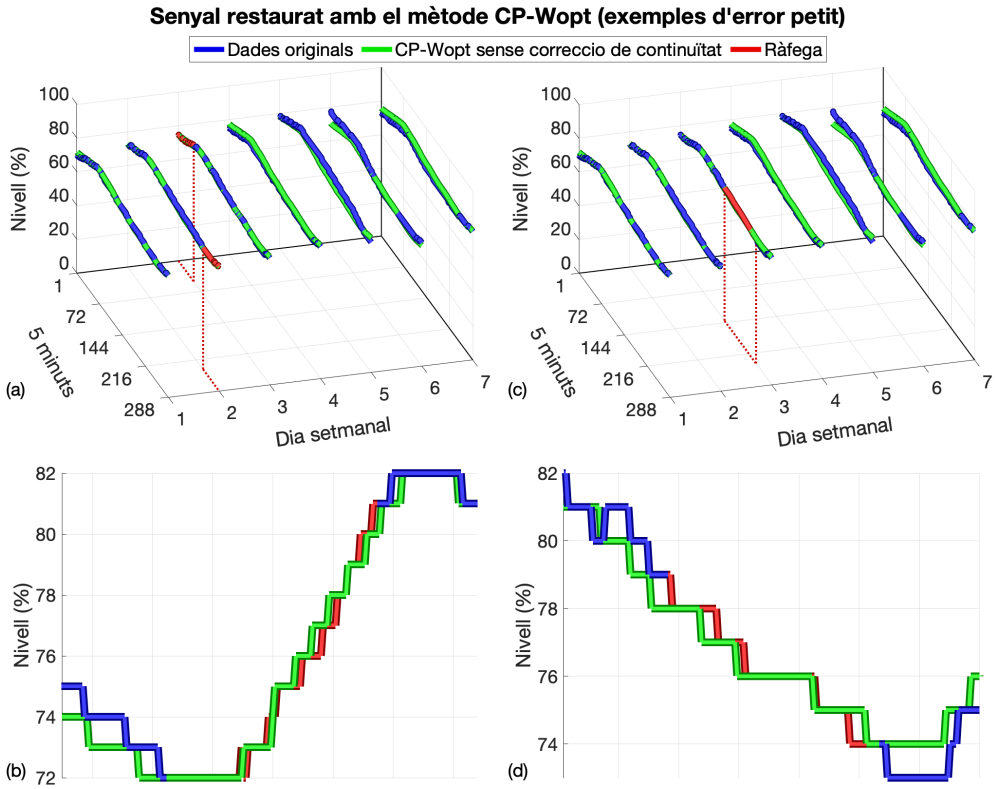

Figura 5.2: Reconstrucció de ràfegues perdudes del sensor de nivell del dipòsit de Castell d'en Planes amb l'algoritme CP-Wopt. Exemples d'error petit amb un tensor $\chi^{288\times7\times3}$. En blau trobem el senyal original, en vermell les dades eliminades i en cian el resultat de la restauració amb CP-Wopt. Es pot observar com el senyal restaurat segueix força bé la tendència del senyal original, tot i que hi ha dies o trams amb més error.

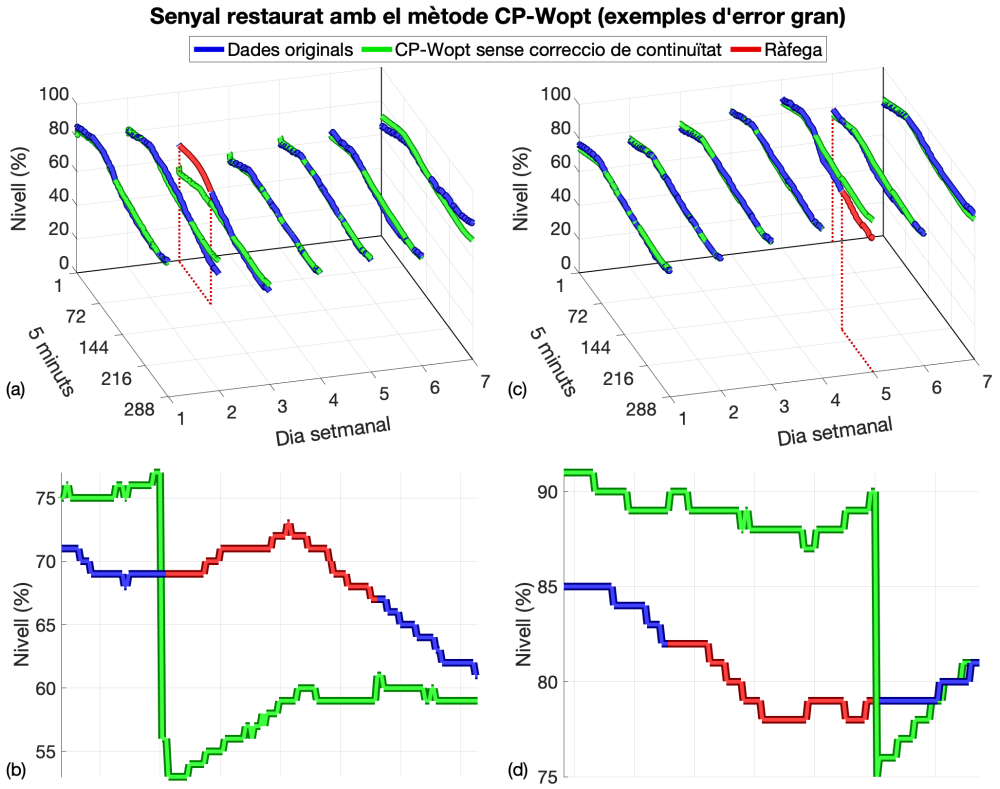

Figura 5.3: Reconstrucció de ràfegues perdudes del sensor de nivell del dipòsit de Castell d'en Planes amb l'algoritme CP-Wopt. Exemples d'error gran amb un tensor $\chi^{288\times7\times3}$. En blau trobem el senyal original, en vermell les dades eliminades i en cian el resultat de la restauració amb CP-Wopt. Es pot observar com el senyal restaurat segueix força bé la tendència del senyal original, tot i que hi ha dies o trams amb més error.

5.2. Descripció del mètode proposat

5.2.1. Introducció de les dades al tensor

La introducció d'un vector de dades, tensor d'una sola dimensió, a un tensor amb més dimensions, permet organitzar les dades de forma que es puguin percebre relacions que d'una altra forma serien més difícils de veure. En el cas presentat es crea un tensor de 3 dimensions temporals χ de mida $288 \times 7 \times n_w$ de la següent manera: una dimensió representa l'hora del dia, l'altra el dia de la setmana i la última el número de setmana de l'historial de dades total. Per fer-ho es fa servir el primer index del tensor per indicar l'hora del dia. Com que la freqüència de mostreig és de 5 minuts i això suposen 288 mostres diàries, la primera dimensió es configura amb 288 mostres, referents a cada una de les mostres d'un dia concret que s'indica amb el segon índex. De forma

que el segon índex indica el dia de la setmana, 1-7 de dilluns a diumenge respectivament. Finalment el terce índex, n_w, indica el número de setmanes de l'historial de dades, que es fan servir per omplir el tensor.

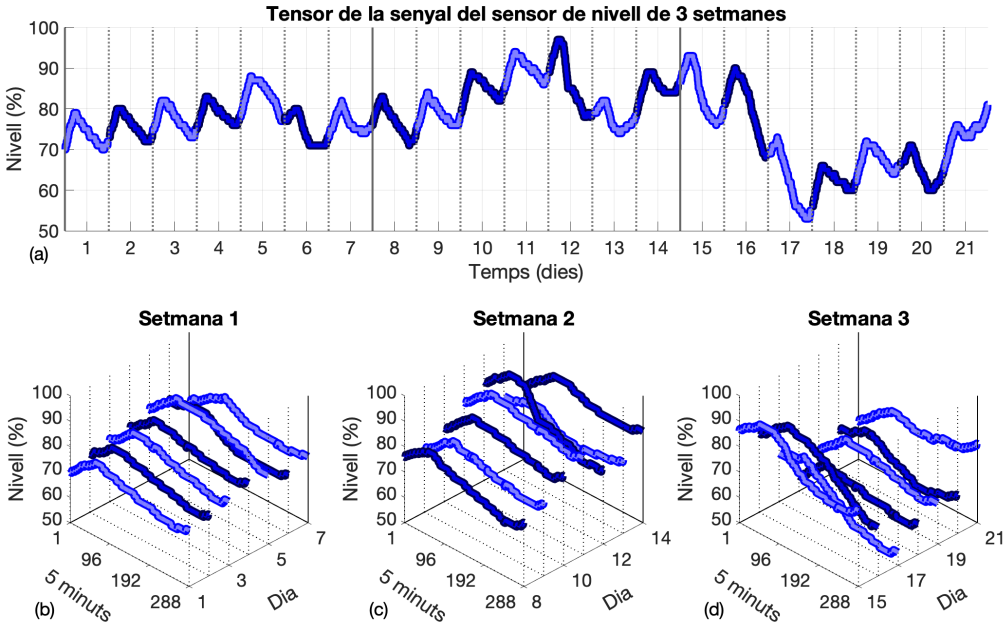

Figura 5.4: Representació de la transformació d'un vector **x**, corresponent a les mesures de tres setmanes, a un tensor $\chi^{288\times7\times3}$. (a) representa el senyal **x**. Les línies verticals primes mostren la separació diària i les gruixudes la setmanal. (b) mostra la primera setmana del tensor, $\chi(:,:,1)$, on cada vector $\chi(:,i,1)$, $i \in [1,7]$ són les mesures diàries del dia i. (c) i (d) mostren la mateixa representació per la segona setmana $\chi(:,:,2)$ i la tercera $\chi(:,:,3)$, respectivament.

En la figura 5.4 es mostra aquest procés. La figura 5.4 (a) mostra l'evolució del senyal de nivell durant tres setmanes. Es pot veure la baixa resolució del sensor amb l'efecte escalonat del senyal. Tot i que no s'aprecia una forma específica del senyal per cada dia, si que sembla haver-hi algunes similituds entre ells. Les figura 5.4 (b, c i d) mostra el tensor $\chi^{288\times7\times3}$ de mida $288\times7\times3$. Tenint en compte que 5.4(b) correspon a les dades de la primera setmana $\chi(:,:,1)$, mentre 5.4(c) i 5.4(d) corresponen a la segona i la tercera respectivament.

Segons aquesta notació, $\chi(:,:,1)$, $\chi(:,:,2)$ i $\chi(:,:,3)$ són matrius, i $\chi(:,1,1)$ és un vector de 288 mostres corresponent al dilluns de la primera setmana. Del tensor $\chi^{288\times7\times3}$ es pot obtenir el senyal original amb l'operació $vec(\boldsymbol{X}_{(1)})$, on $\boldsymbol{X}_{(1)}$ és la matriu en mode-1 de χ.

5.2.2. Procediment

S'utilitza el tensor proposat a l'apartat *Introducció de les dades al tensor* per mirar d'aprofitar els possibles patrons diaris i setmanals que es puguin trobar en el senyal de nivell. Es fan servir les descomposicions descrites a l'apartat *Models Tucker i CANDECOMP/PARAFAC* per estimar les fluctuacions del senyal i aprofitar-les per restaurar les dades perdudes, comparant els resultats obtinguts amb cada un dels dos models més comuns.

Degut a què les descomposicions tensorials que es volen fer servir no permeten treballar amb tensors que tenen dades perdudes o buides, el primer pas és omplir la ràfega de dades perduda amb l'estimació realitzada per un mètode lineal. Un cop es disposa d'un tensor sense posicions buides, s'aplica una de les descomposicions seleccionades: Tucker o CP. Amb la selecció d'una determinada descomposició, es limiten els modes a cada una de les dimensions, i per tant es simplifica el tensor, de forma que al re-composar el tensor de nou es filtren les irregularitats incorporades pel mètode lineal. Després d'aquest procés es disposa d'una aproximació del senyal de nivell, que no és exacte, però segueix amb força fidelitat les fluctuacions del senyal original.

5.2.3. Correcció de la continuïtat

Per solucionar el problema del desplaçament que pateix sovint el senyal recuperat de la descomposició tensorial, mencionat a l'apartat *Algoritme CP-Wopt*, s'aplica un procés per no perdre la continuïtat del senyal i adaptar les fluctuacions capturades pel tensor a les posicions de la ràfega de dades perdudes, figures 5.5-5.13. Es fa de forma semblant al procés realitzat en l'apartat *Combinació de FIR endavant i enrere* de la secció 4.3 on es combinen diferents versions del predictor de Wiener.

Considerant que x són les dades originals, $\hat{x}$ són les mostres obtingudes amb el procés de descomposició del tensor i que es realitza la restauració d'una ràfega de mida N mostres (que comença a la posició i_1 i acaba a la posició i_N), s'estableixen dos "offsets" per corregir el desplaçament de les mostres: l'inicial O_1 i el final O_N. O_1 correspon al desplaçament que s'observa en la mostra anterior a la ràfega de x, és a dir $x(i_1-1)$, respecte a la de $\hat{x}$, és a dir $\hat{x}(i_1-1)$. O_N correspon, de forma recíproca, al desplaçament de la mostra posterior a la

ràfega, $(i_N + 1)$.

$$\begin{cases} O_1 = x(i_1 - 1) - \hat{x}(i_1 - 1) \\ O_N = x(i_N + 1) - \hat{x}(i_N + 1) \end{cases} \tag{5.4}$$

Llavors si $\tilde{x}$ són les mostres restaurades, $\tilde{x} = x$, excepte per les mostres corresponents a la ràfega, les posicions $i = i_1 \dots i_N$, en que s'aplica la correcció de continuïtat mitjançant la següent expressió:

$$\text{per} \quad i = i_1 \dots i_N; \qquad \tilde{x}(i) = \begin{cases} \hat{x}(i) - \dfrac{O_1 + O_N}{2} & \text{if } N = 1 \\ \hat{x}(i) - \dfrac{(N-i)O_1 + (i-1)O_N}{N-1} & \text{if } N > 1 \end{cases} \tag{5.5}$$

A continuació a la figures 5.5-5.13 es mostren varis exemples.

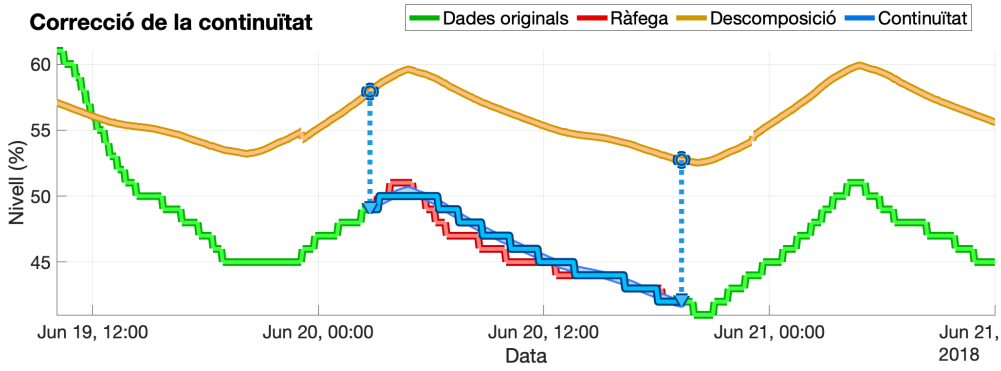

Figura 5.5: La línia verda és el senyal original, x_i; la vermella les 200 mostres perdudes; la taronja el resultat de la descomposició Tucker(1,1,1), $\hat{x}_i$; la blava l'efecte de la correcció, $\tilde{x}_i$; i les fletxes els "offsets" inicial i final, O_0 i O_{B+1}.

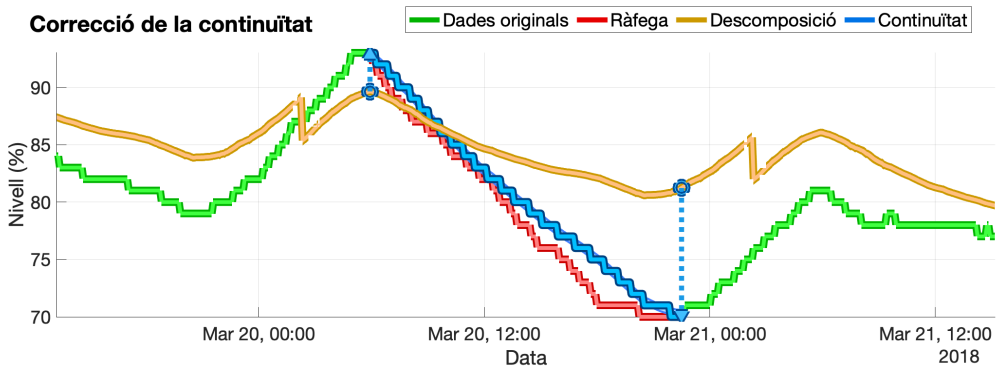

Figura 5.6: La línia verda és el senyal original, x_i; la vermella les 200 mostres perdudes; la taronja el resultat de la descomposició Tucker(1,1,1), $\hat{x}_i$; la blava l'efecte de la correcció, $\tilde{x}_i$; i les fletxes els "offsets" inicial i final, O_0 i O_{B+1}.

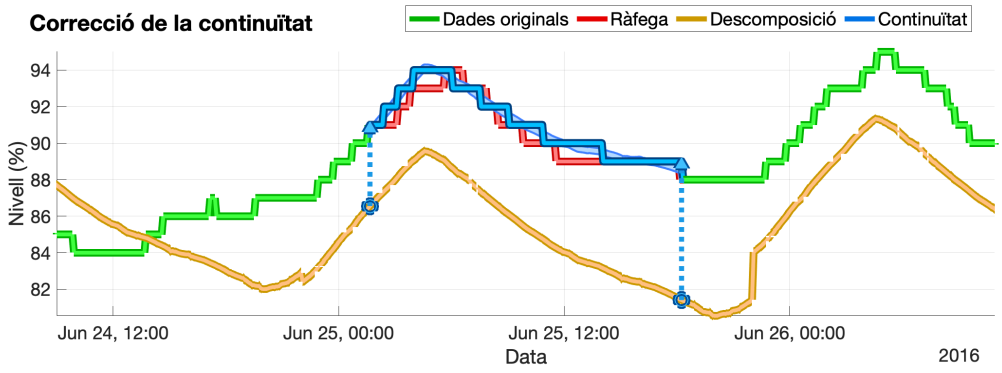

Figura 5.7: La línia verda és el senyal original, x_i; la vermella les 200 mostres perdudes; la taronja el resultat de la descomposició Tucker(1,1,1), $\hat{x}_i$; la blava l'efecte de la correcció, $\tilde{x}_i$; i les fletxes els "offsets" inicial i final, O_0 i O_{B+1}.

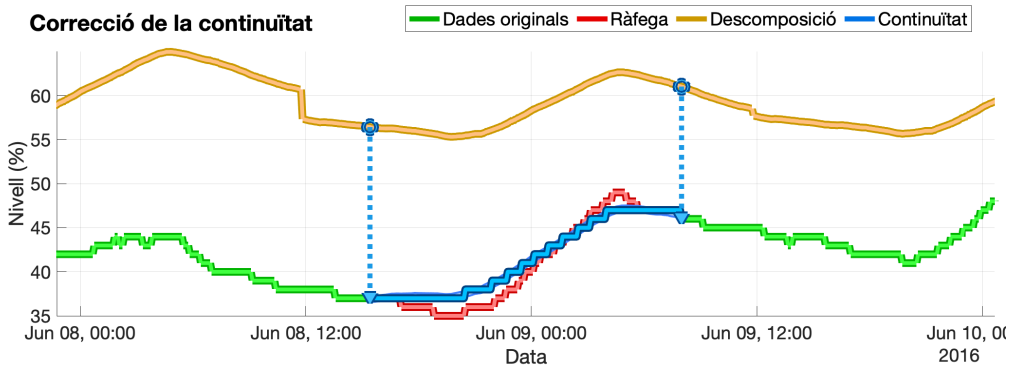

Figura 5.8: La línia verda és el senyal original, x_i; la vermella les 200 mostres perdudes; la taronja el resultat de la descomposició Tucker(1,1,1), $\hat{x}_i$; la blava l'efecte de la correcció, $\tilde{x}_i$; i les fletxes els "offsets" inicial i final, O_0 i O_{B+1}. .

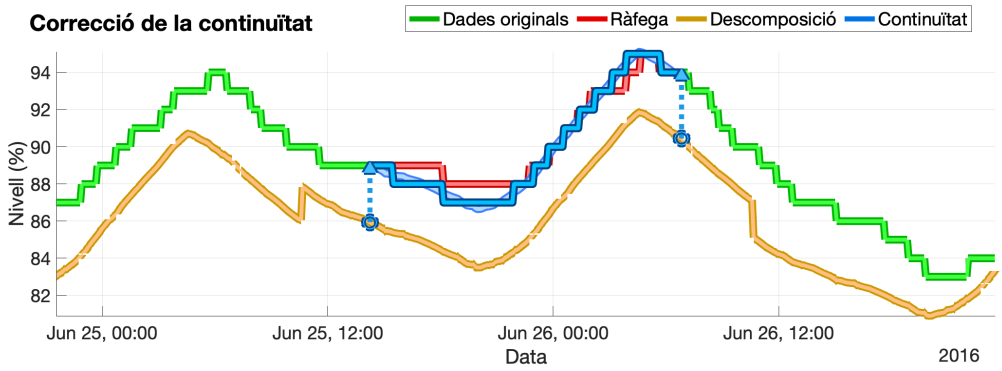

Figura 5.9: La línia verda és el senyal original, x_i; la vermella les 200 mostres perdudes; la taronja el resultat de la descomposició Tucker(1,1,1), $\hat{x}_i$; la blava l'efecte de la correcció, $\tilde{x}_i$; i les fletxes els "offsets" inicial i final, O_0 i O_{B+1}. .

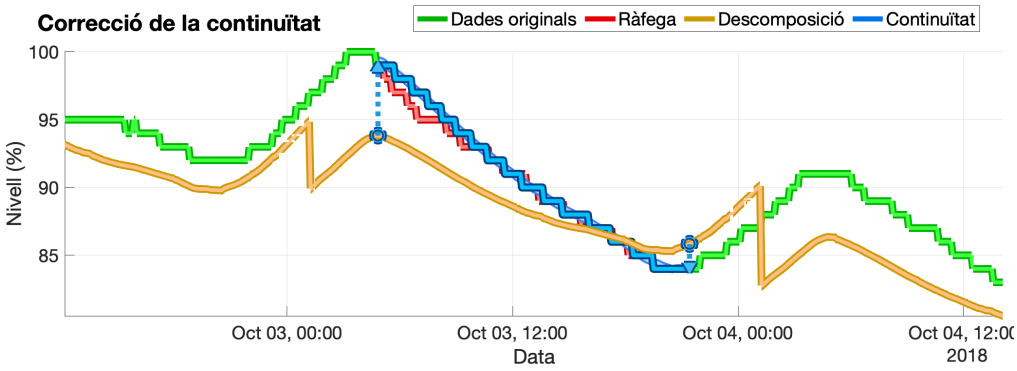

Figura 5.10: La línia verda és el senyal original, x_i; la vermella les 200 mostres perdudes; la taronja el resultat de la descomposició Tucker(1,1,1), $\hat{x}_i$; la blava l'efecte de la correcció, $\tilde{x}_i$; i les fletxes els "offsets" inicial i final, O_0 i O_{B+1}. .

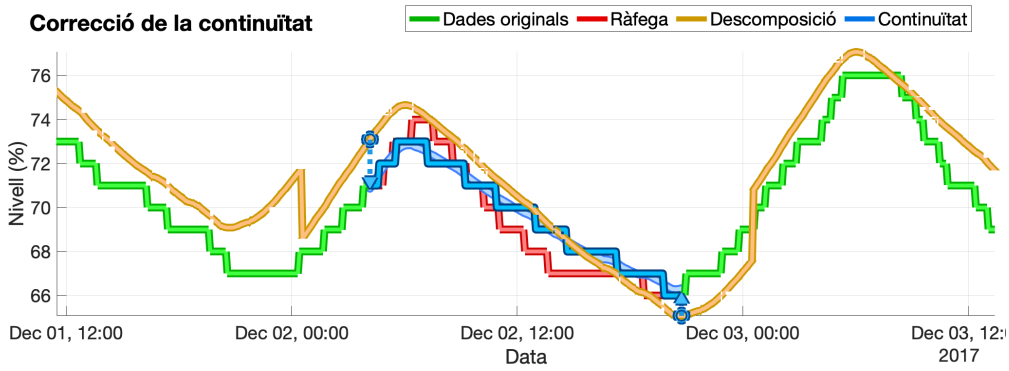

Figura 5.11: La línia verda és el senyal original, x_i; la vermella les 200 mostres perdudes; la taronja el resultat de la descomposició Tucker(1,1,1), $\hat{x}_i$; la blava l'efecte de la correcció, $\tilde{x}_i$; i les fletxes els "offsets" inicial i final, O_0 i O_{B+1}. .

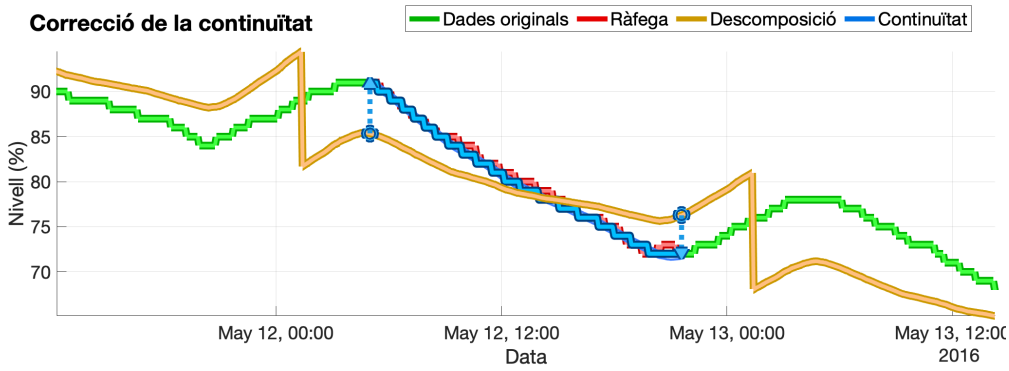

Figura 5.12: La línia verda és el senyal original, x_i; la vermella les 200 mostres perdudes; la taronja el resultat de la descomposició Tucker(1,1,1), $\hat{x}_i$; la blava l'efecte de la correcció, $\tilde{x}_i$; i les fletxes els "offsets" inicial i final, O_0 i O_{B+1}. .

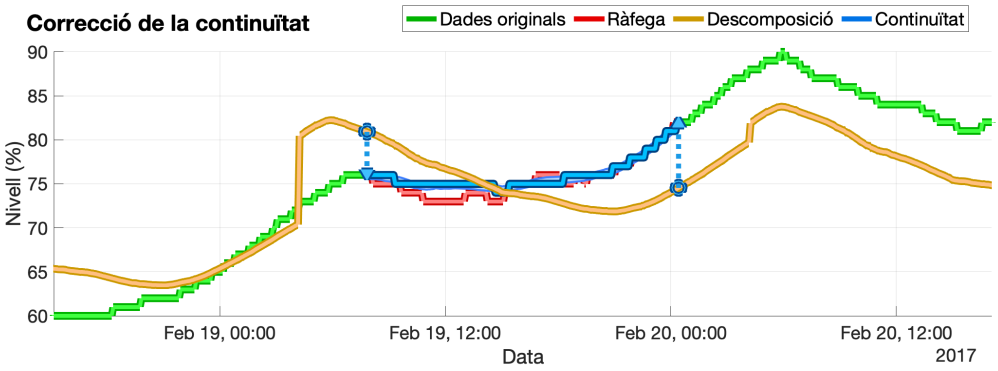

Figura 5.13: La línia verda és el senyal original, x_i; la vermella les 200 mostres perdudes; la taronja el resultat de la descomposició Tucker(1,1,1), $\hat{x}_i$; la blava l'efecte de la correcció, $\tilde{x}_i$; i les fletxes els "offsets" inicial i final, O_0 i O_{B+1}. .

5.3. Configuració del mètode proposat

5.3.1. Nucli de la descomposició

Per tal d'optimitzar el mètode proposat es realitzen vàries simulacions per cada un dels models seleccionats, Tucker i CP, per tal de determinar la configuració òptima del nucli de la descomposició. Es fa servir el tensor $\chi^{288\times7\times3}$ i rafegues de 100 i 200 mostres per fer aquesta prova. Com a mètodes lineals es proven els mètodes dels operaris de les seccions 4.1 i 4.2 i el mètode FIR (L/M) amb les configuracions seleccionades a l'apartat *Configuració de L i M* de la mateixa secció. Per cada un dels casos es realitza una simulació aplicant i no aplicant la correcció de continuïtat de l'apartat *Correcció de la continuïtat*, per avaluar-ne l'efecte. Com a referències per comparar resultats, també es mostren els errors al fer servir només el mètode lineal i el mètode CP-Wopt (que és independent del mètode lineal utilitzat ja que treballa amb valors buits en el tensor original, i per tant no necessita com a primera fase omplir les dades buides del tensor amb un mètode lineal).

Pel cas del model Tucker es proven diferents configuracions pel nucli $G^{L\times M\times N}$, provant L de 1 a 3, M de 1 a 7 (ja que és l'índex referent al dia de la setmana) i N de 1 a 3 (que seria la mida màxima per al tensor $\chi^{288\times7\times3}$). Els resultats obtinguts en aquesta prova es mostren a la figures 5.14 i 5.15. La millor configuració per tots els mètodes lineals provats és utilitzar el nucli $G^{3\times3\times1}$, excepte pel mètode "Dada anterior", de la secció 4.1, i només en el cas de 200 mostres de ràfega.

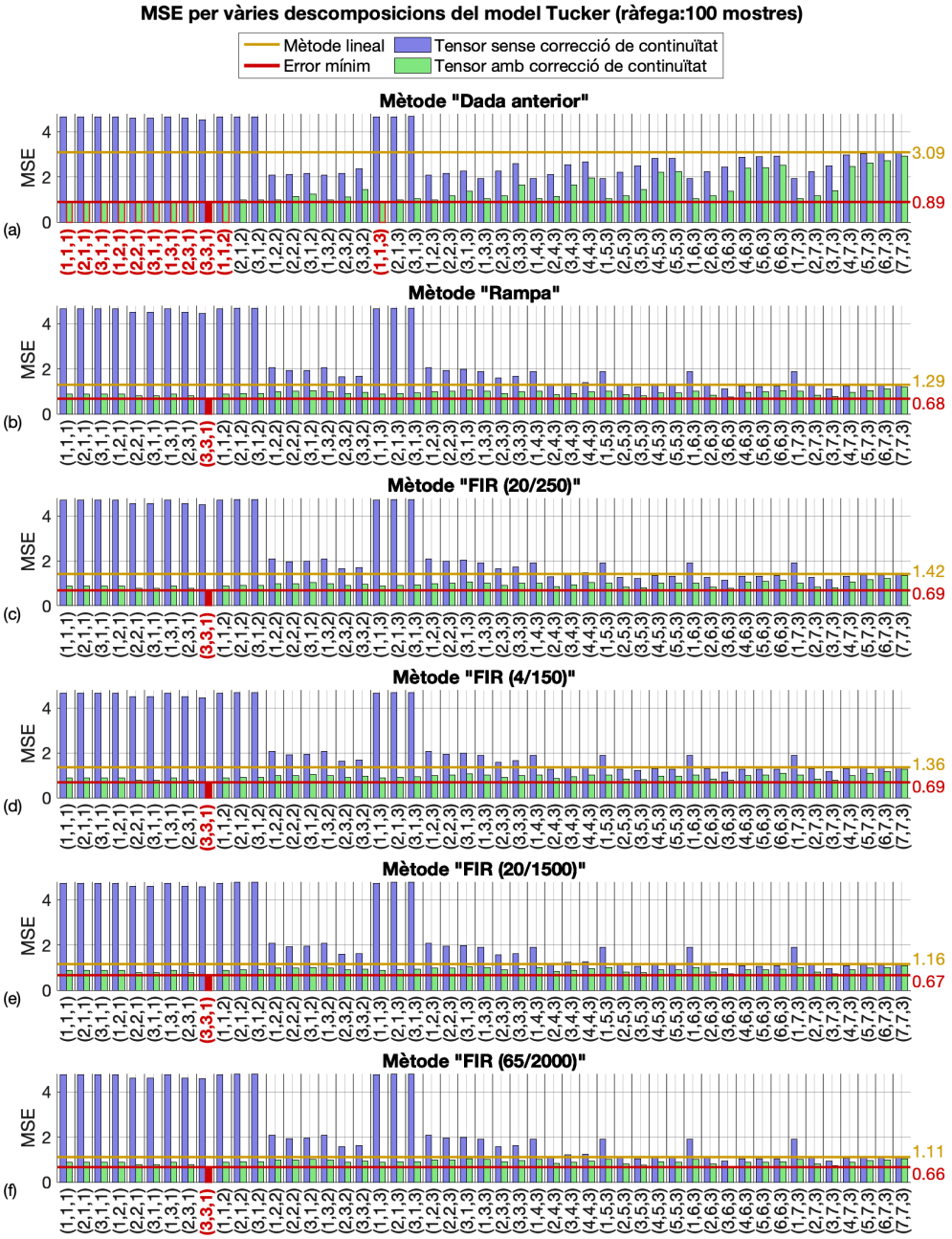

Figura 5.14: MSE segons el nucli de la descomposició Tucker amb un tensor $\chi^{288\times7\times3}$ i 100 mostres de ràfega i per varis mètodes lineals (i algunes configuracions del mètode "FIR"). Es mostra l'MSE aplicant i no aplicant correcció de continuïtat. Les barres marcades en vermell indiquen un error com a molt un 5 % més gran que el mínim.

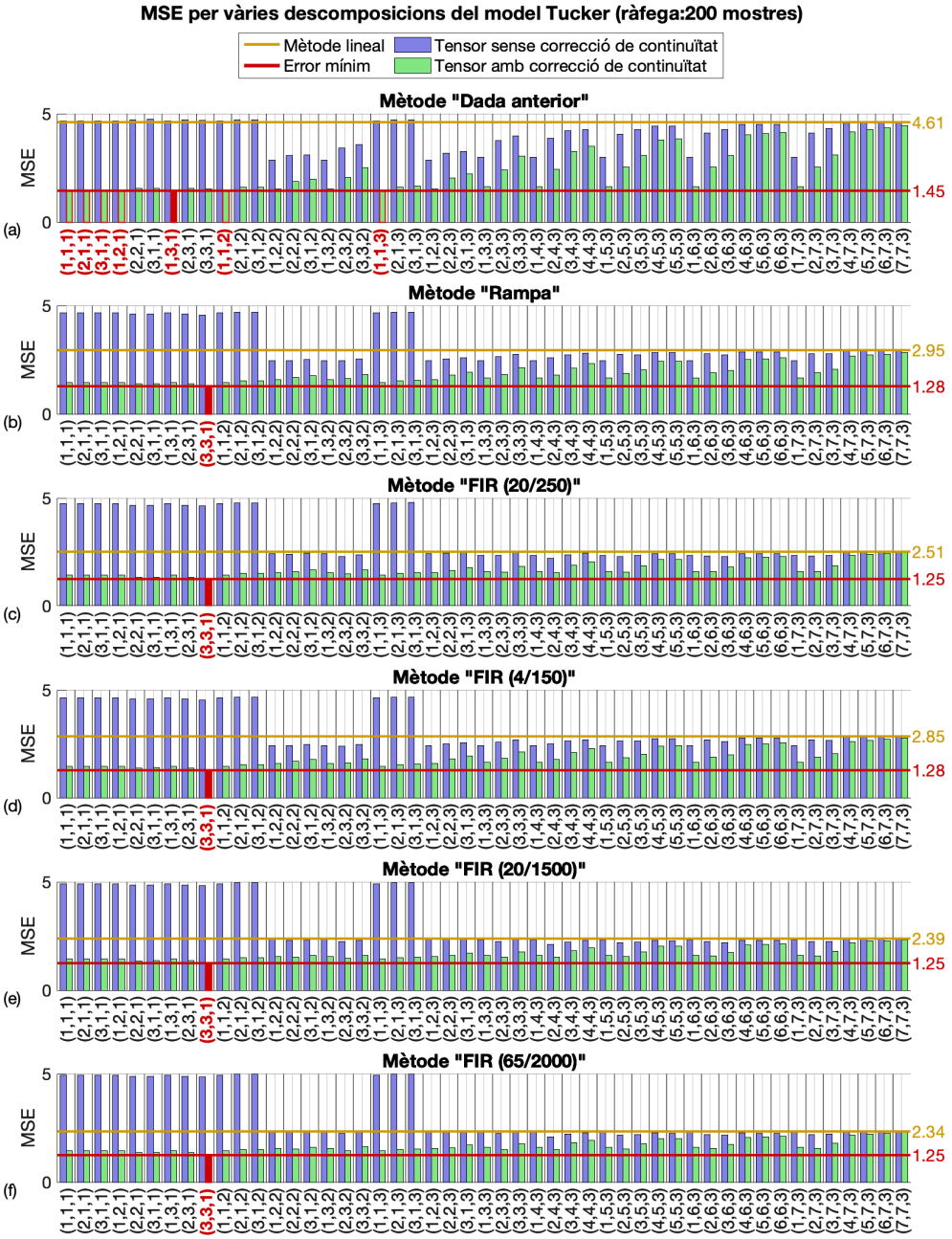

Figura 5.15: MSE segons el nucli de la descomposició Tucker amb un tensor $\chi^{288\times7\times3}$ i 200 mostres de ràfega i per varis mètodes lineals (i algunes configuracions del mètode "FIR"). Es mostra l'MSE aplicant i no aplicant correcció de continuïtat. Les barres marcades en vermell indiquen un error com a molt un 5 % més gran que el mínim.

Pel al model CP es proven diferents configuracions pel nucli $G^{D\times D\times D}$, provant D de 2 a 15, ja que el cas D=1 és equivalent al cas Tucker amb nucli $G^{1\times 1\times 1}$. Els resultats es mostren a la figures 5.16 i 5.17. La millor configuració pels casos de 100 mostres de ràfega és el nucli $G^{6\times 6\times 6}$, excepte en el mètode "Dada anterior" que és $G^{2\times 2\times 2}$ (semblant al que passa amb el model Tucker). Per als casos de 200 mostres de ràfega la millor opció pel nucli és $G^{2\times 2\times 2}$ en tots els mètodes.

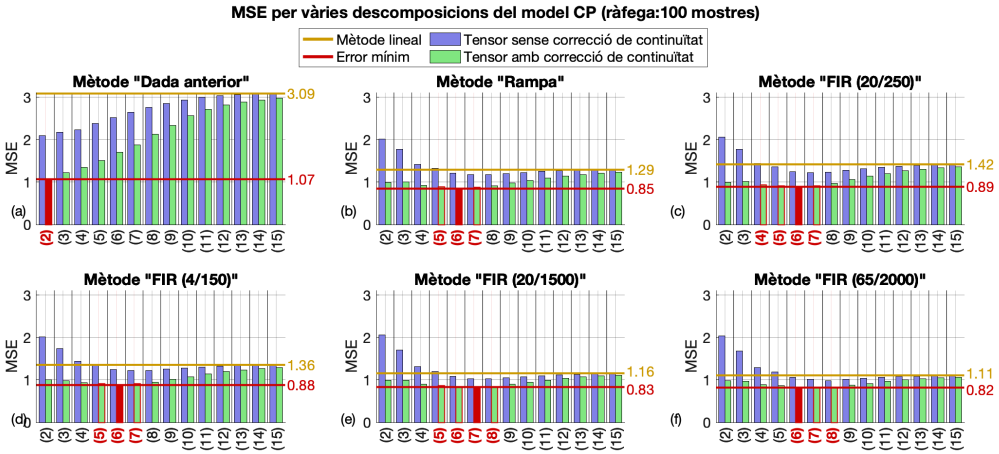

Figura 5.16: MSE segons el nucli de la descomposició CP amb un tensor $\chi^{288\times 7\times 3}$ i 100 mostres de ràfega i per varis mètodes lineals (i algunes configuracions del mètode "FIR"). Es mostra l'MSE aplicant i no aplicant correcció de continuïtat. Les barres marcades en vermell indiquen un error com a molt un 5 % més gran que el mínim.

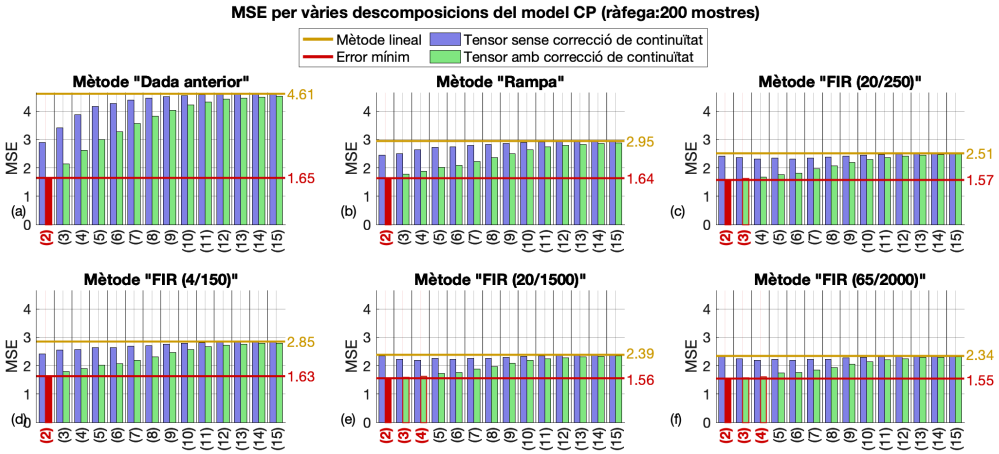

Figura 5.17: MSE segons el nucli de la descomposició CP amb un tensor $\chi^{288\times 7\times 3}$ i 200 mostres de ràfega i per varis mètodes lineals (i algunes configuracions del mètode "FIR"). Es mostra l'MSE aplicant i no aplicant correcció de continuïtat. Les barres marcades en vermell indiquen un error com a molt un 5 % més gran que el mínim.

A les figures 5.18-5.21 es mostren els resultats per les configuracions que han resultat més efectives en la prova realitzada. En concret es seleccionen: Tuker(2,2,1), Tuker(3,3,1), CP(2) i CP(6). També es s'inclou la Tucker(1,1,1) ja que es considera interessant al ser equivalent a la CP(1), i al ser la descomposició més simple possible. Com en el cas anterior també es mostren els errors del mètode lineal i de l'algoritme CP-Wopt per poder realitzar la comparació entre mètodes. El resultats indiquen que la millor configuració és la TK(3,3,1), excepte en el cas del mètode lineal anomenat "Dada anterior" (secció 4.1) amb tensor de 3 setmanes, llavors la millor és TK(1,1,1) o CP(1).

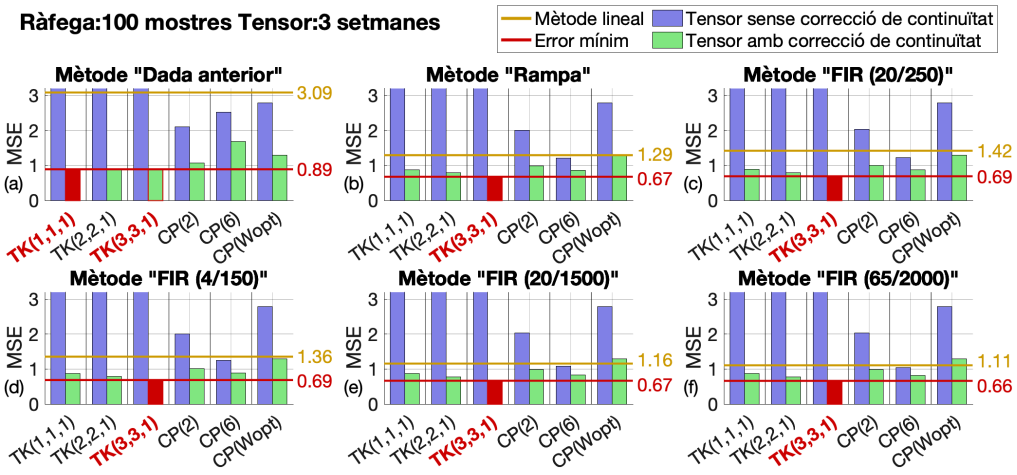

Figura 5.18: MSE obtingut amb un tensor $\chi^{288\times7\times3}$ i una ràfega de 100 mostres. L'MSE vermell és com a molt un 5 % més gran que el mínim.

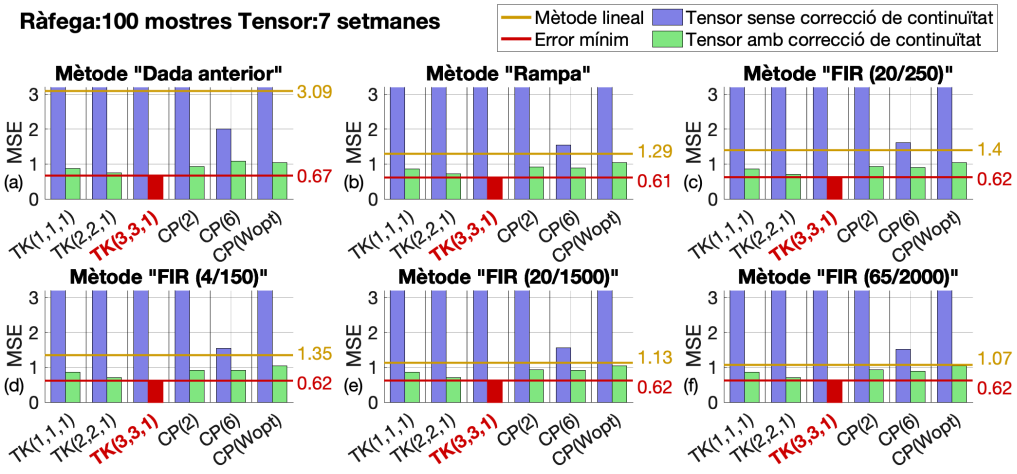

Figura 5.19: MSE obtingut amb un tensor $\chi^{288\times7\times3}$ i una ràfega de 100 mostres. L'MSE vermell és com a molt un 5 % més gran que el mínim.

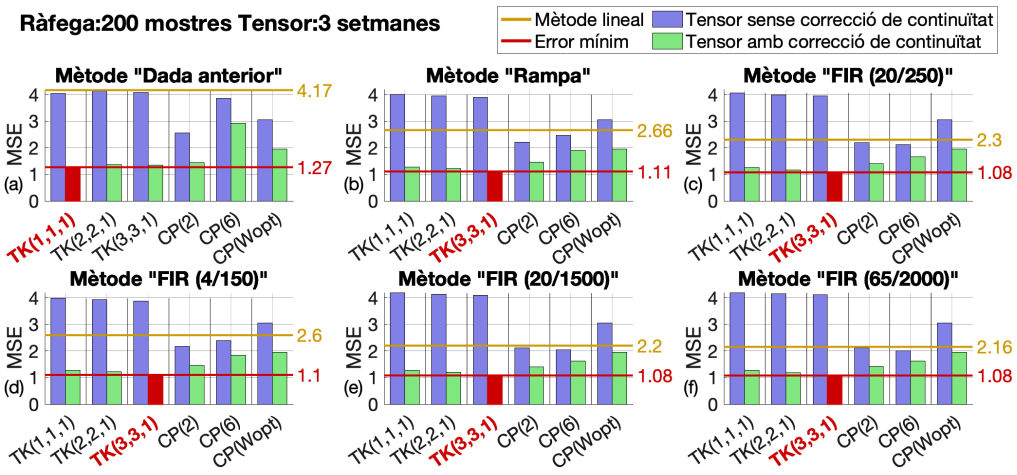

Figura 5.20: MSE obtingut amb un tensor $\chi^{288\times7\times3}$ i una ràfega de 100 mostres. L'MSE vermell és com a molt un 5 % més gran que el mínim.

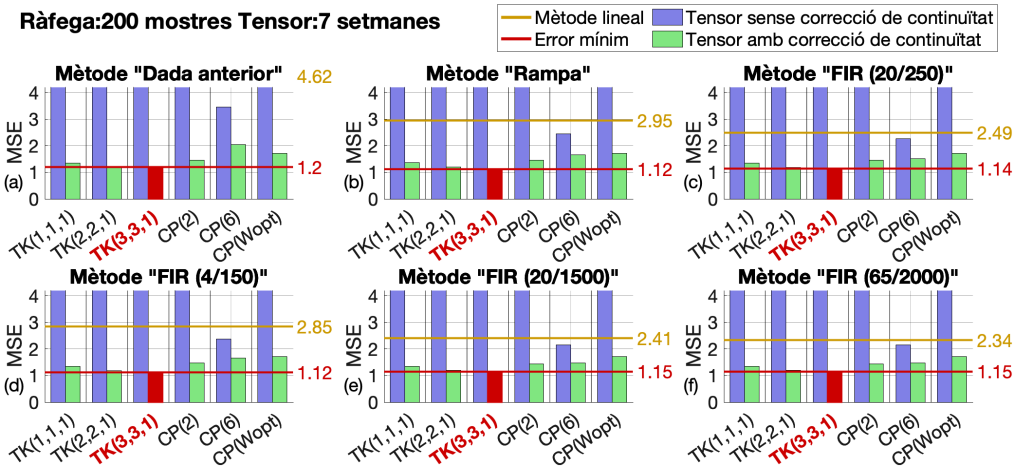

Figura 5.21: MSE obtingut amb un tensor $\chi^{288\times7\times3}$ i una ràfega de 100 mostres. L'MSE vermell és com a molt un 5 % més gran que el mínim.

5.3.2. Mida del tensor

Per tal d'avaluar l'efecte de la mida del tensor a partir de l'MSE es realitzen un conjunt de simulacions mantenint la mida de la ràfega a 100 mostres i augmentant la mida del tensor utilitzat de 3 a 31 setmanes. El nombre de setmanes utilitzades sempre és senar degut a que es força que la setmana on es localitza la ràfega sigui la del centre del tensor, és a dir, sempre s'agafen el mateix nombre de setmanes anteriors i posteriors a la setmana on es troba la ràfega.

Per tant, la configuració del tensor en aquest experiment compleix $\chi^{288\times7\times2i+1}$,

on amb $i = 1$ s'obté el tensor més simple, el de 3 setmanes utilitzat fins ara en les simulacions $\chi^{288\times7\times3}$, i amb $i = 15$ el més gran $\chi^{288\times7\times31}$. Els resultats es poden veure a continuació a la figura 5.22.

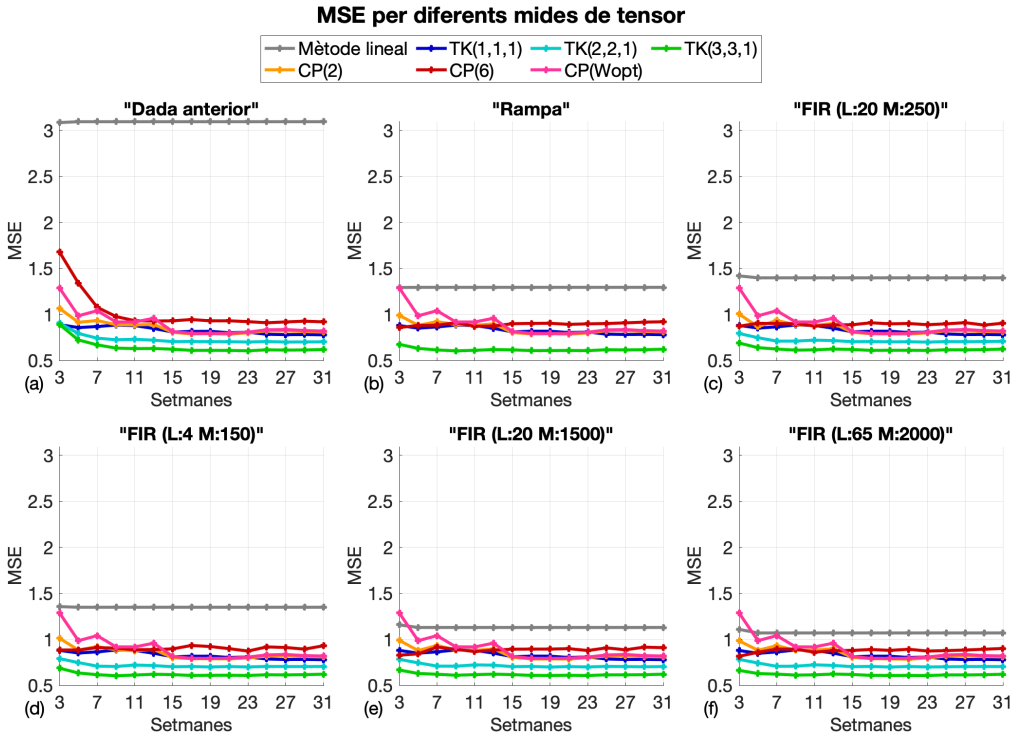

Figura 5.22: Gràfic de la tendència del MSE segons la mida del tensor, és a dir, segons el nombre de setmanes de l'historial de dades que es fan servir per omplir-lo. La ràfega restaurada és de 100 mostres.

5.3.3. Mida de la ràfega

De forma similar al apartat anterior, també es calcula l'MSE obtingut mantenint la mida del tensor i modificant el número de mostres de la ràfega. Per fer-ho es realitzen un conjunt de simulacions amb el tensor $\chi^{288\times7\times3}$, i es proven ràfegues des de 5 mostres de longitud fins a 250 mostres. La ràfega de 250 mostres representaria quasi un dia sencer de dades perdudes, cas poc probable ja que els operaris o els serveis tècnics no solen tardar tant a reparar la fallada de les comunicacions o d'un sensor important del sistema SCADA. Els resultats obtinguts en aquest experiment es poden veure a continuació a la figura 5.23.

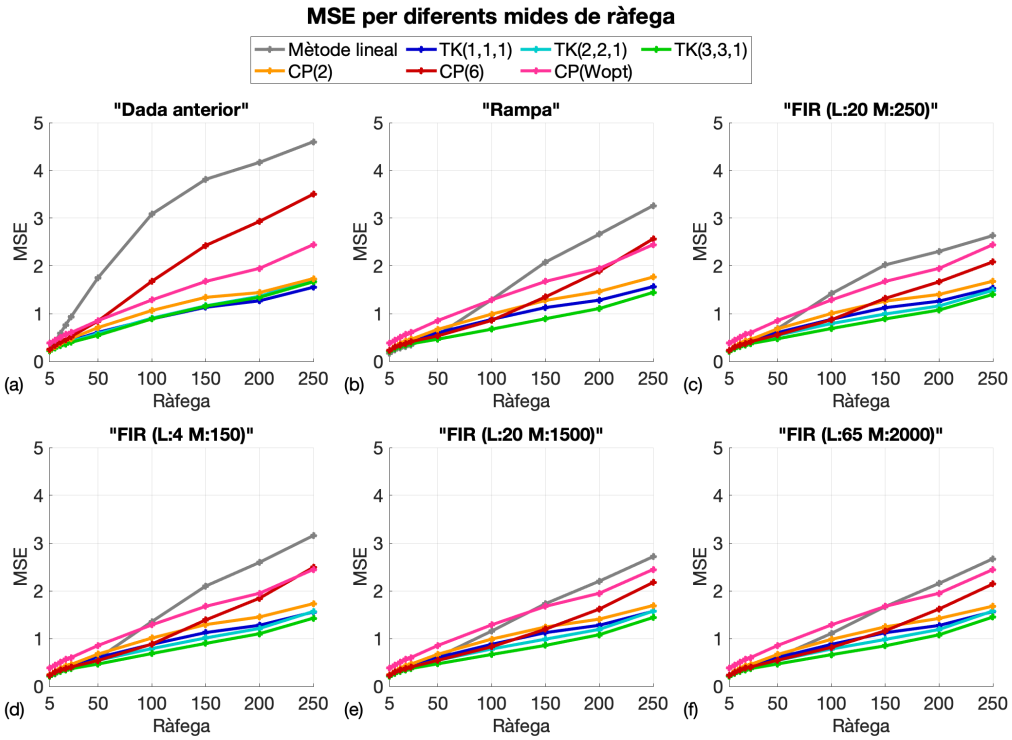

Figura 5.23: Gràfic de la tendència del MSE segons la mida de la ràfega, és a dir, segons el nombre de mostres perdudes consecutivament. El test es realitza amb un tensor $\chi^{288\times7\times3}$, de tres setmanes.

5.4. Resultats

Els mètodes basats en tensors s'han aplicat amb èxit en àrees com la mineria de dades [42] o el processament de senyal [8] i recentment també en el camp de la reconstrucció de dades perdudes amb bons resultat, fent que apareguin noves estratègies. Algunes d'elles deriven de les millors i més conegudes tècniques de descomposició com a [47, 48] amb la t-SVD (tensor-Single Value Decomposition), a [10] amb la Tucker, o a [11, 12] amb la CANDECOMP/PARAFAC (CP). D'altres menys conegudes utilitzen estratègies diferents com l'optimització de Riemannian, [49]. En termes generals existeixen dues vies per realitzar la restauració de dades a partir dels tensors, la que es coneix com a "maximization of expectations", que implica omplir els valors buits abans de realitzar el procediment tensorial i és la que farem servir amb la metodologia proposada. L'altra es coneix com a "marginalization", i es tracta d'ignorar els valors perduts durant el procés d'estimació. Un dels mètodes que fan servir "marginalization" es el CP-Wopt (CP Weighted OPTimization) [46]. Aquest algoritme ve d'aplicar el model CANDECOMP/PARAFAC (CP) en el cas concret de considerar que

falten dades, i replantejant els pesos de les mostres per tal de no tenir en compte les posicions amb mostres buides. La versió disponible en Matlab d'aquest algoritme es fa servir per comparar-ne els resultats amb l'algoritme desenvolupat utilitzant els models Tucker i CP.

L'àlgebra tensorial permet explotar la relació entre diferents dimensions en que s'organitzen les dades en el tensor. Per exemple l'algoritme CP-Wopt és capaç de recuperar les components correctes, a partir de dades sorolloses amb fins a un 99 % de dades perdudes quan altres mètodes col·lapsen només amb un 25-40 % [46]. Segons les característiques de les dades, les condicions de les pèrdues, la seva distribució estadística i l'aplicació particular, la recuperació de dades es pot millorar amb solucions fetes a mida, com en el cas d'aquesta tesi o l'exemple mostrat a [50]. Com s'ha comentat hi ha varis models de descomposició de tensors, però els que s'han considerat en aquesta tesi són dos dels models més coneguts i usats, el CANDECOMP/PARAFAC (CP) [11, 12] i el Tucker [10, 39]. Aquests models s'han utilitzat en l'algorisme desenvolupat en aquest capítol que fa l'àlgebra tensorial per tal de refinar les reconstruccions de dades realitzades amb mètode lineal convencionals. A continuació es mostren varis exemples dels resultats per les configuracions TK(3,3,1) i CP(2) en les figures 5.24-5.42.

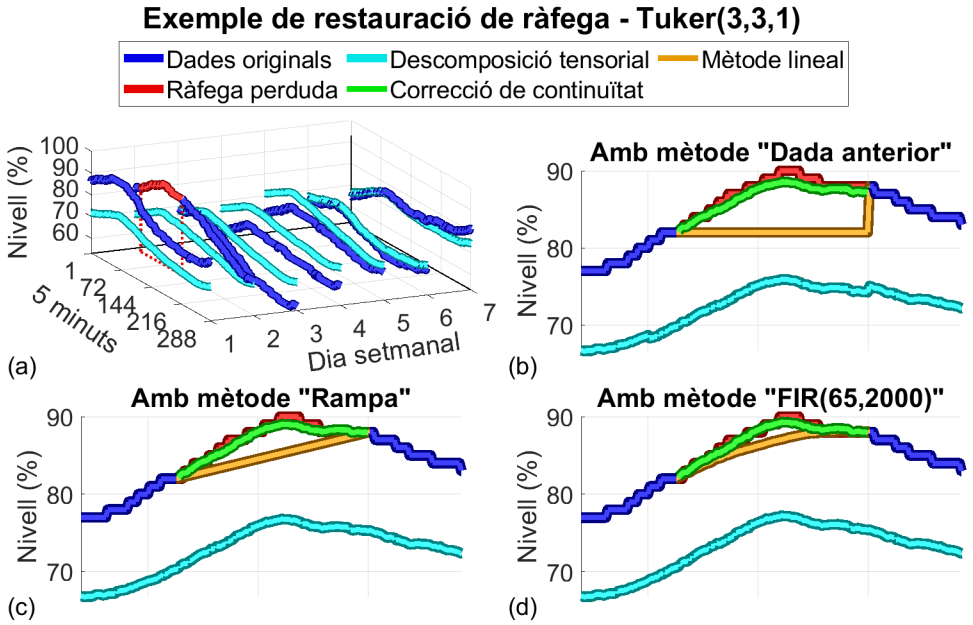

Figura 5.24: a) gràfic 3D del senyal original i de la descomposició Tuker(3,1,1) durant la setmana on es troba la ràfega. b), c, i d) resultat del tensor i de la correcció de la continuïtat per a cada un dels mètodes lineals descrits.

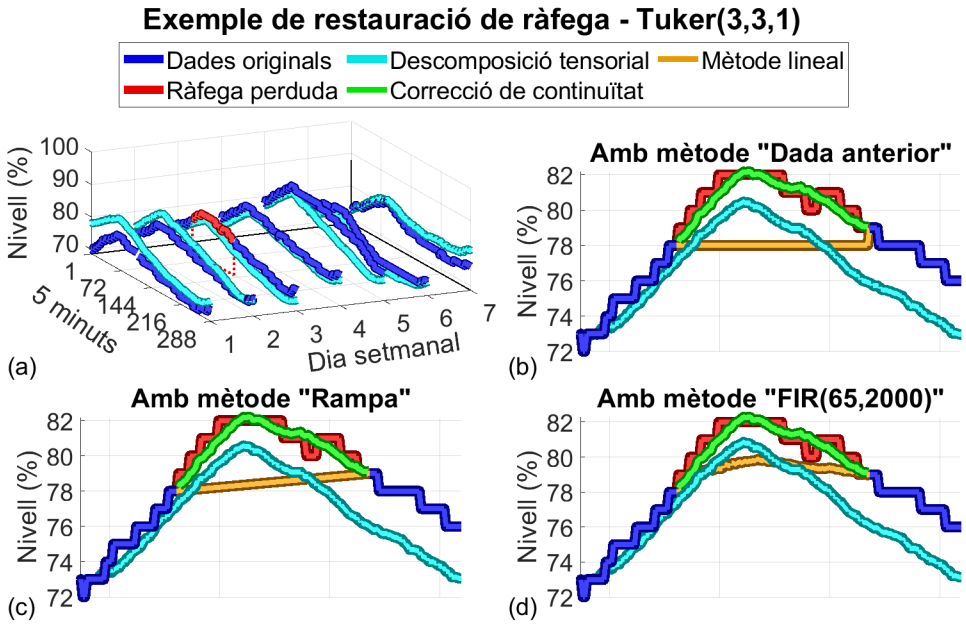

Figura 5.25: a) gràfic 3D del senyal original i de la descomposició Tuker(3,1,1) durant la setmana on es troba la ràfega. b), c, i d) resultat del tensor i de la correcció de la continuïtat per a cada un dels mètodes lineals descrits.

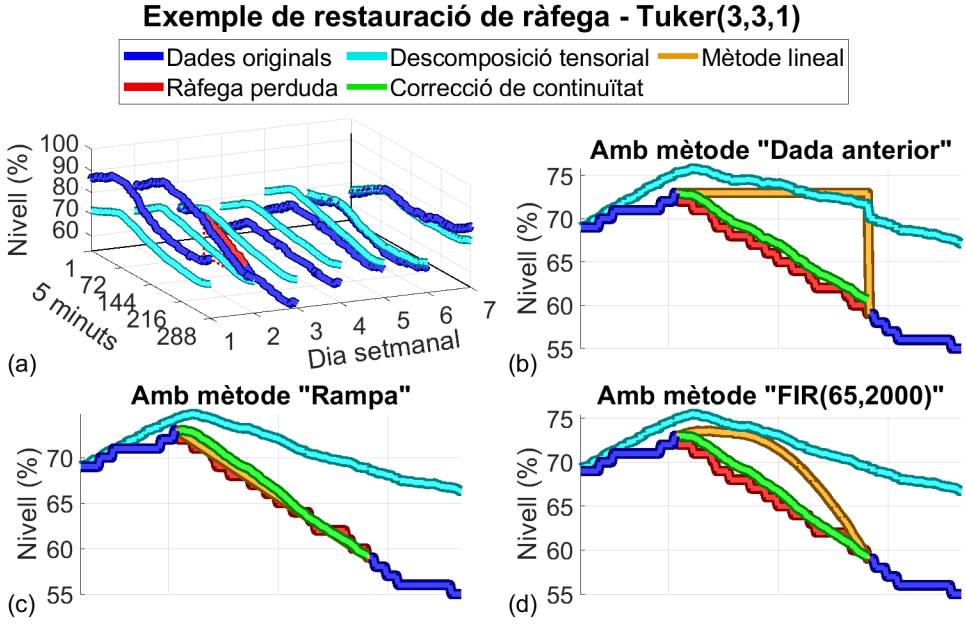

Figura 5.26: a) gràfic 3D del senyal original i de la descomposició Tuker(3,1,1) durant la setmana on es troba la ràfega. b), c, i d) resultat del tensor i de la correcció de la continuïtat per a cada un dels mètodes lineals descrits.

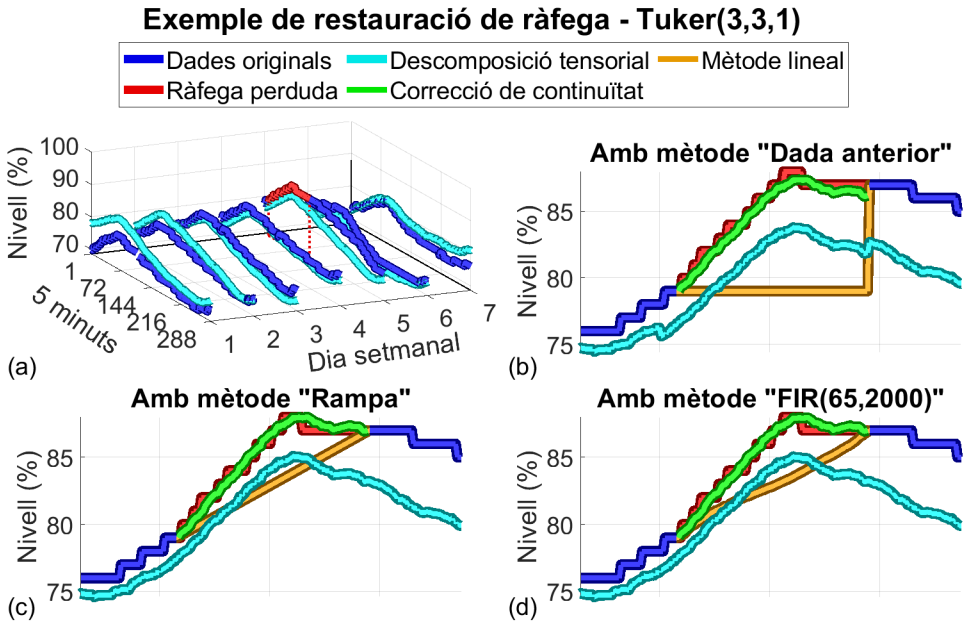

Figura 5.27: a) gràfic 3D del senyal original i de la descomposició Tuker(3,1,1) durant la setmana on es troba la ràfega. b), c, i d) resultat del tensor i de la correcció de la continuïtat per a cada un dels mètodes lineals descrits.

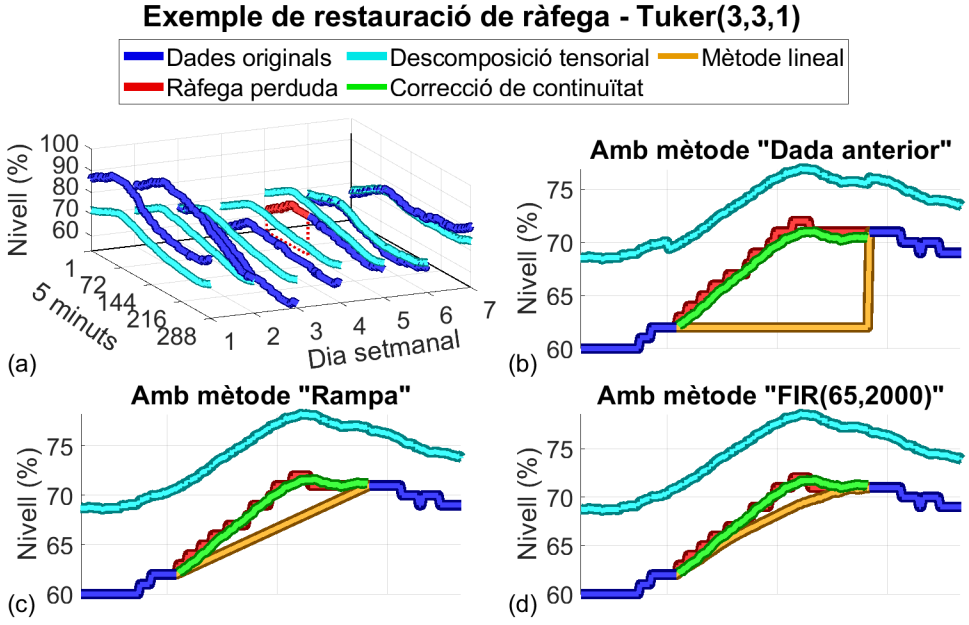

Figura 5.28: a) gràfic 3D del senyal original i de la descomposició Tuker(3,1,1) durant la setmana on es troba la ràfega. b), c, i d) resultat del tensor i de la correcció de la continuïtat per a cada un dels mètodes lineals descrits.

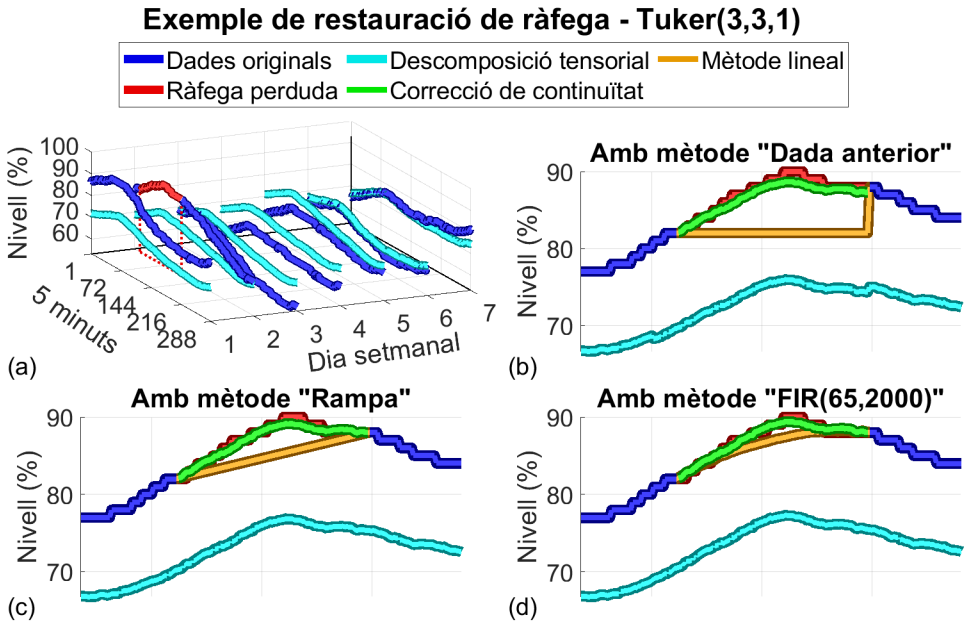

Figura 5.29: a) gràfic 3D del senyal original i de la descomposició Tuker(3,1,1) durant la setmana on es troba la ràfega. b), c, i d) resultat del tensor i de la correcció de la continuïtat per a cada un dels mètodes lineals descrits.

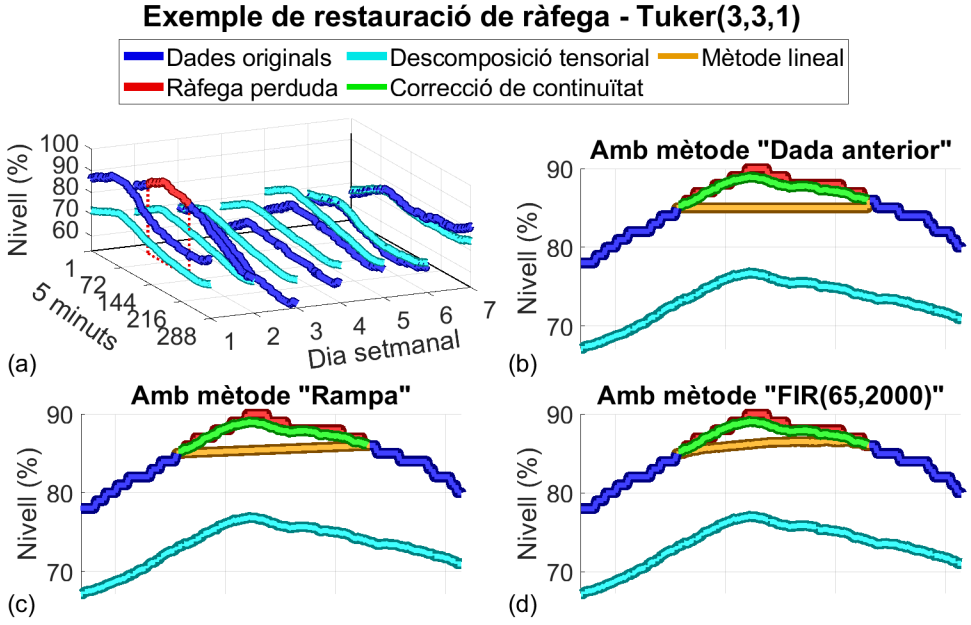

Figura 5.30: a) gràfic 3D del senyal original i de la descomposició Tuker(3,1,1) durant la setmana on es troba la ràfega. b), c, i d) resultat del tensor i de la correcció de la continuïtat per a cada un dels mètodes lineals descrits.

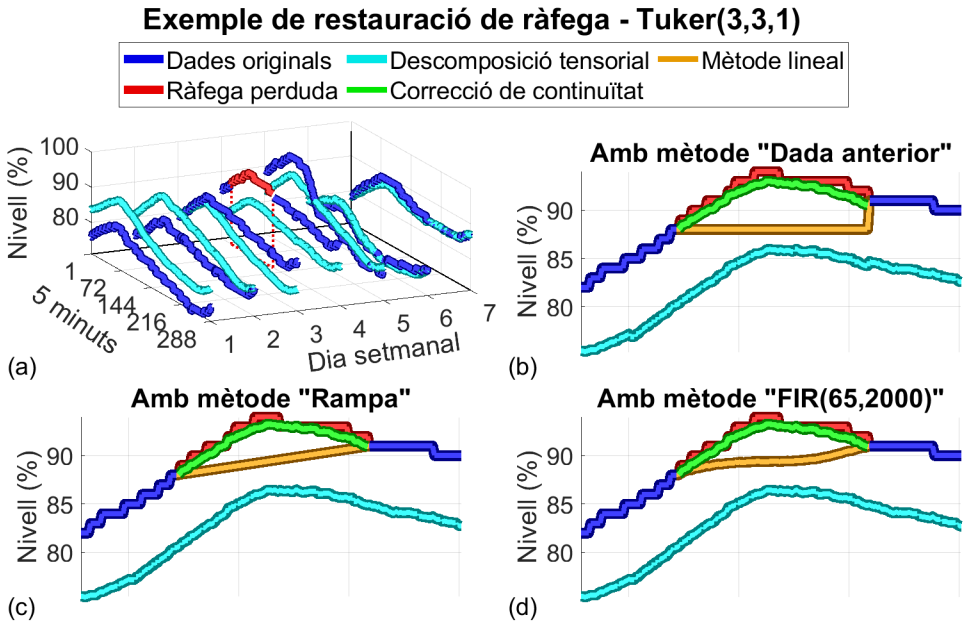

Figura 5.31: a) gràfic 3D del senyal original i de la descomposició Tuker(3,1,1) durant la setmana on es troba la ràfega. b), c, i d) resultat del tensor i de la correcció de la continuïtat per a cada un dels mètodes lineals descrits.

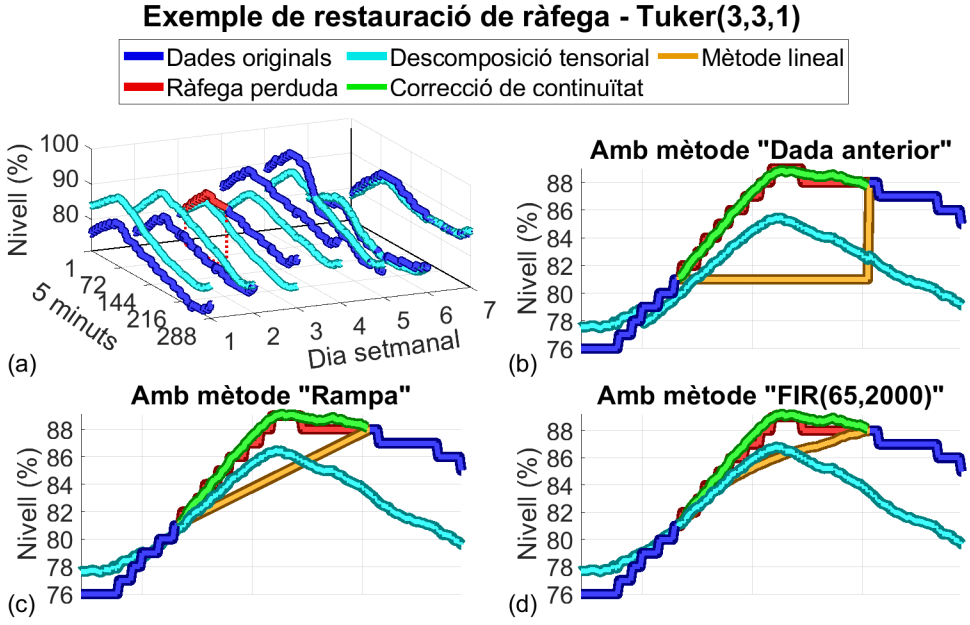

Figura 5.32: a) gràfic 3D del senyal original i de la descomposició Tuker(3,1,1) durant la setmana on es troba la ràfega. b), c, i d) resultat del tensor i de la correcció de la continuïtat per a cada un dels mètodes lineals descrits.

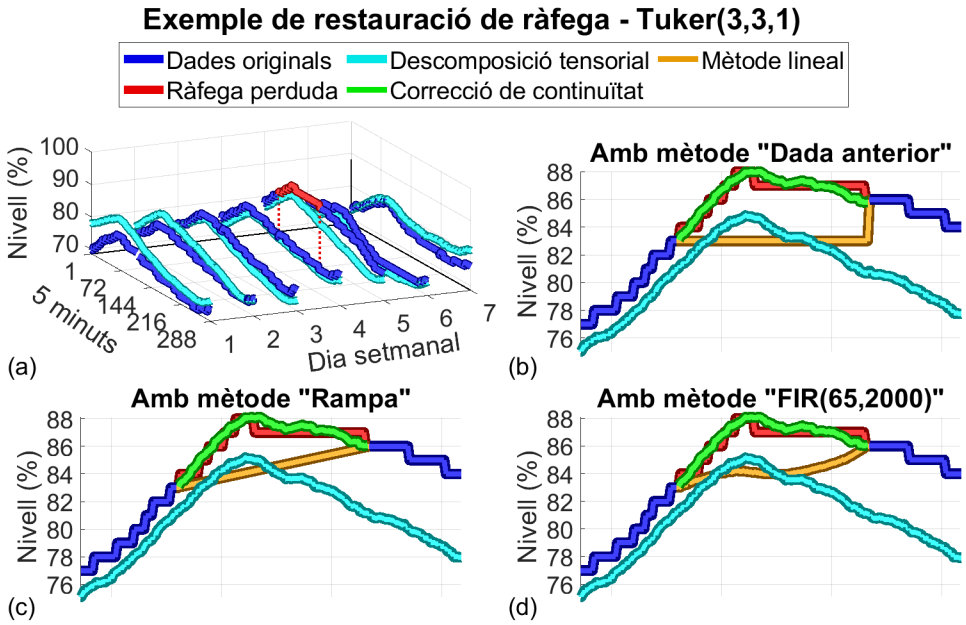

Figura 5.33: a) gràfic 3D del senyal original i de la descomposició Tuker(3,1,1) durant la setmana on es troba la ràfega. b), c, i d) resultat del tensor i de la correcció de la continuïtat per a cada un dels mètodes lineals descrits.

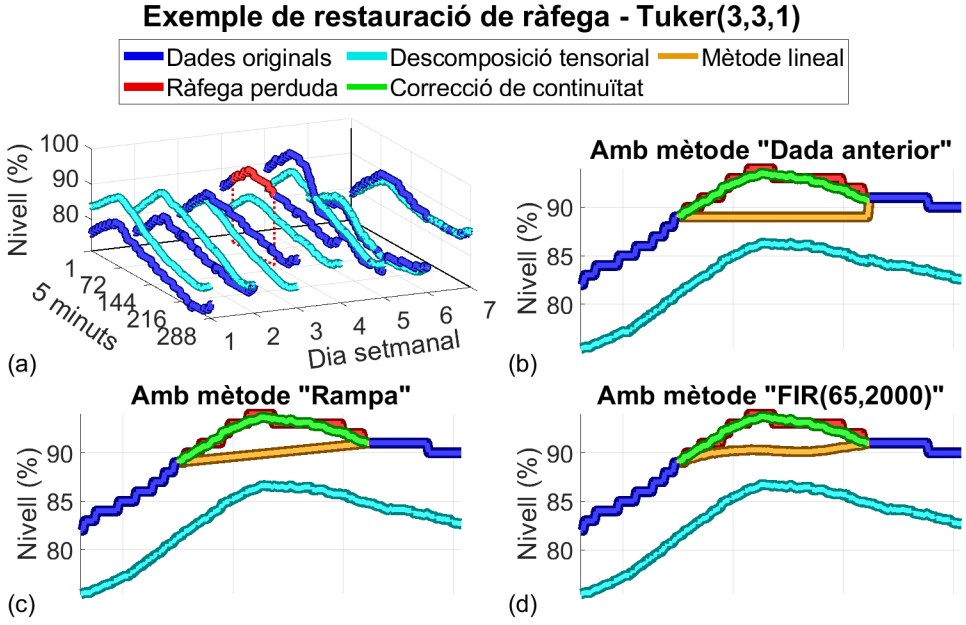

Figura 5.34: a) gràfic 3D del senyal original i de la descomposició Tuker(3,1,1) durant la setmana on es troba la ràfega. b), c, i d) resultat del tensor i de la correcció de la continuïtat per a cada un dels mètodes lineals descrits.

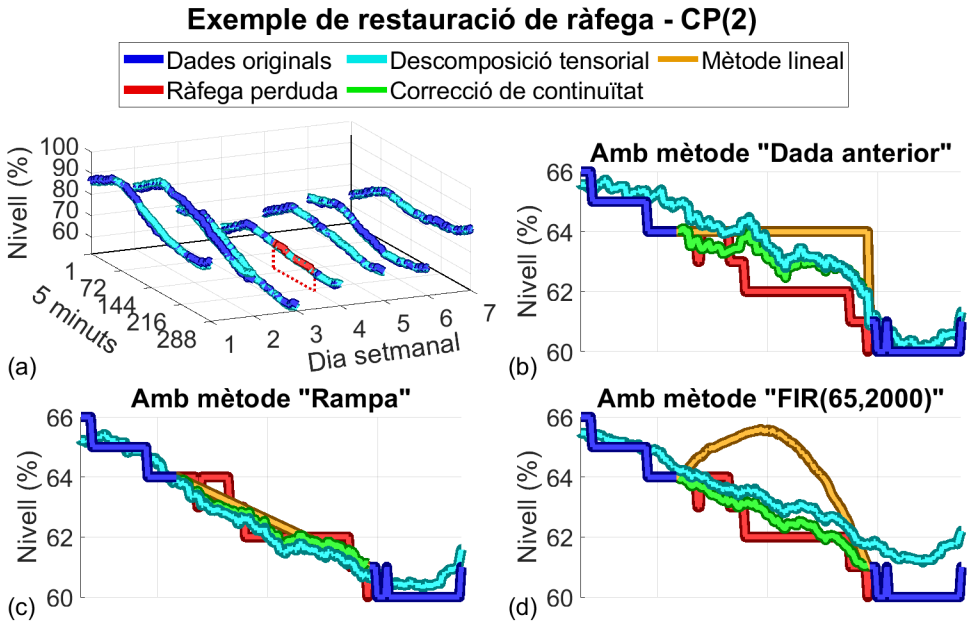

Figura 5.35: a) gràfic 3D del senyal original i de la descomposició CP(2) durant la setmana on es troba la ràfega. b), c, i d) resultat del tensor i de la correcció de la continuïtat per a cada un dels mètodes lineals descrits.

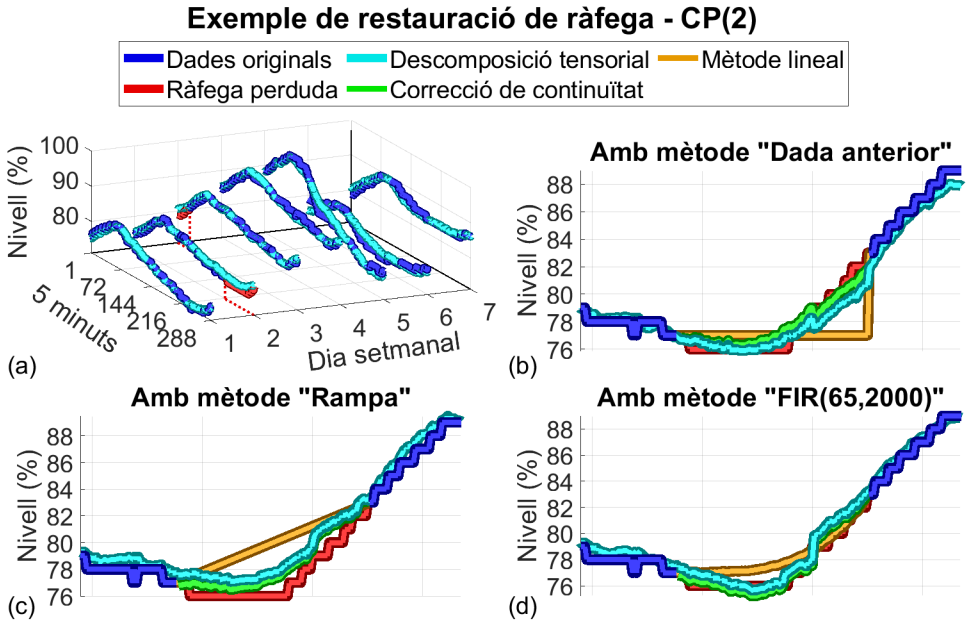

Figura 5.36: a) gràfic 3D del senyal original i de la descomposició CP(2) durant la setmana on es troba la ràfega. b), c, i d) resultat del tensor i de la correcció de la continuïtat per a cada un dels mètodes lineals descrits.

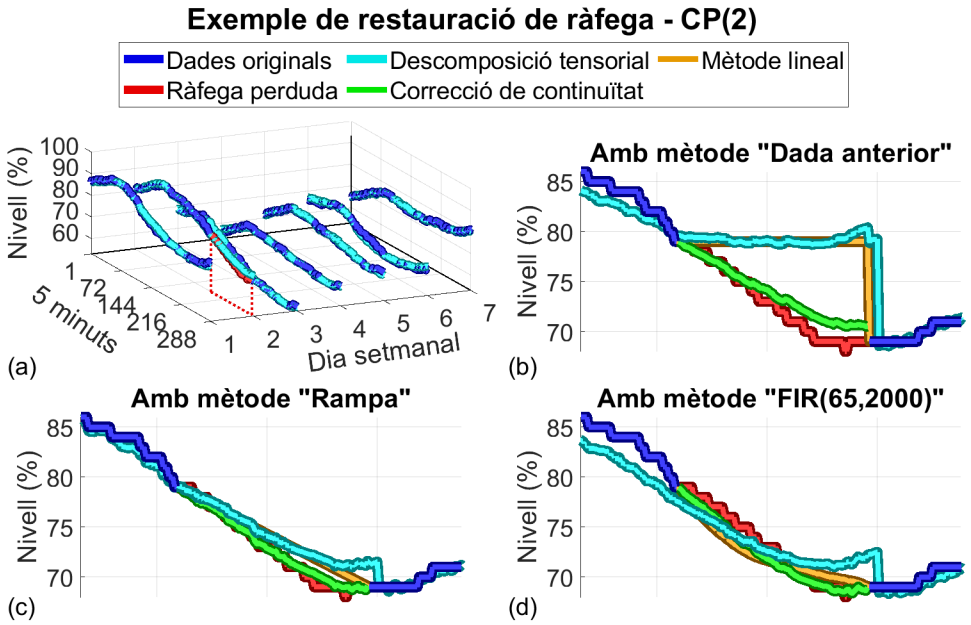

Figura 5.37: a) gràfic 3D del senyal original i de la descomposició CP(2) durant la setmana on es troba la ràfega. b), c, i d) resultat del tensor i de la correcció de la continuïtat per a cada un dels mètodes lineals descrits.

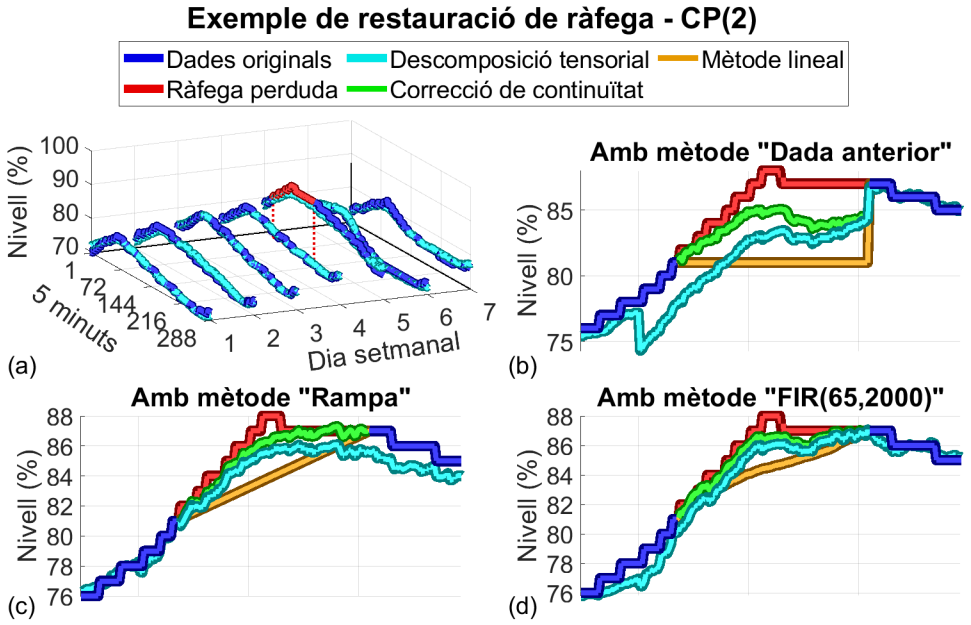

Figura 5.38: a) gràfic 3D del senyal original i de la descomposició CP(2) durant la setmana on es troba la ràfega. b), c, i d) resultat del tensor i de la correcció de la continuïtat per a cada un dels mètodes lineals descrits.

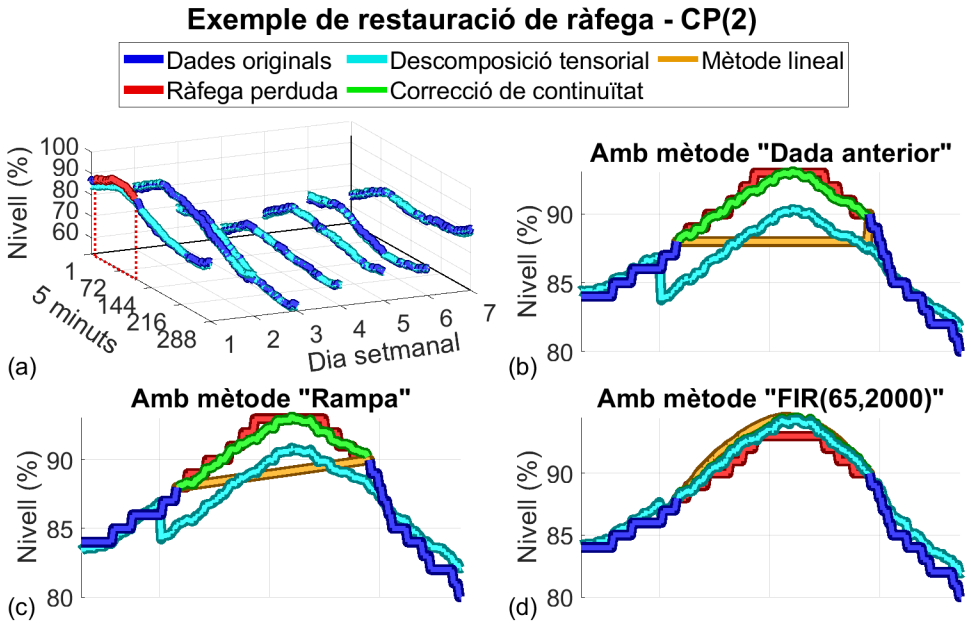

Figura 5.39: a) gràfic 3D del senyal original i de la descomposició CP(2) durant la setmana on es troba la ràfega. b), c, i d) resultat del tensor i de la correcció de la continuïtat per a cada un dels mètodes lineals descrits.

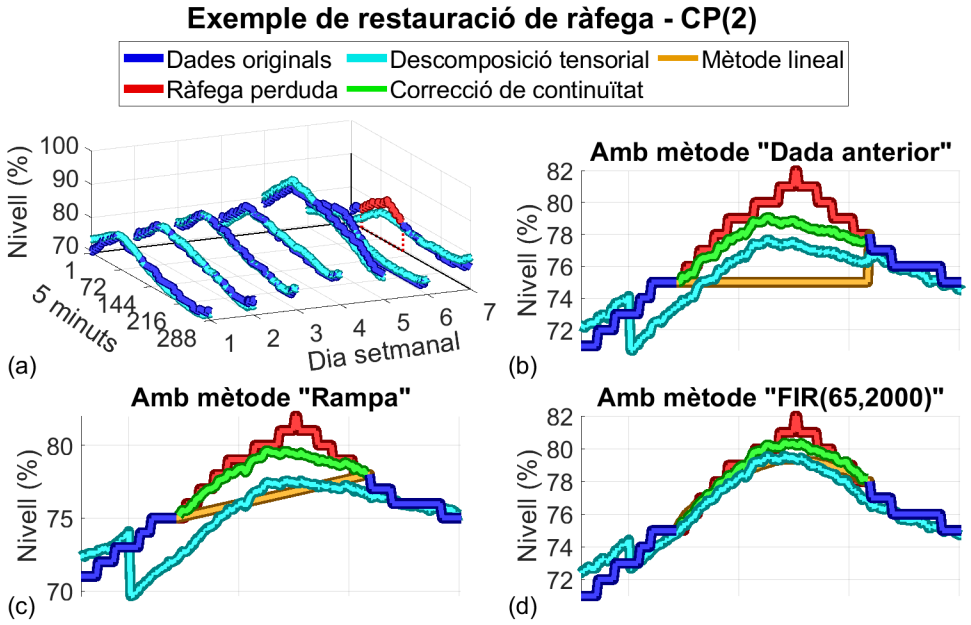

Figura 5.40: a) gràfic 3D del senyal original i de la descomposició CP(2) durant la setmana on es troba la ràfega. b), c, i d) resultat del tensor i de la correcció de la continuïtat per a cada un dels mètodes lineals descrits.

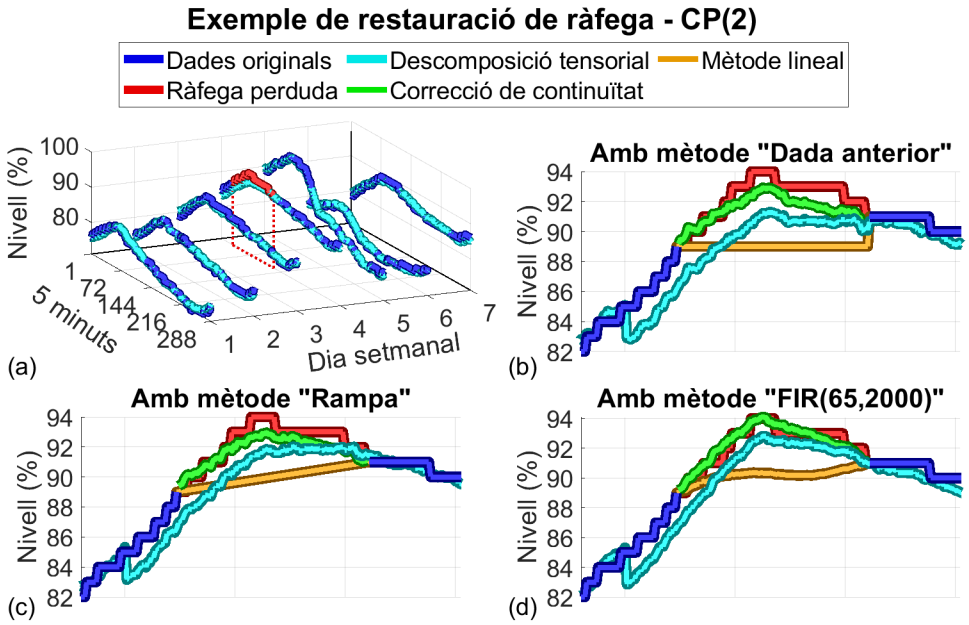

Figura 5.41: a) gràfic 3D del senyal original i de la descomposició CP(2) durant la setmana on es troba la ràfega. b), c, i d) resultat del tensor i de la correcció de la continuïtat per a cada un dels mètodes lineals descrits.

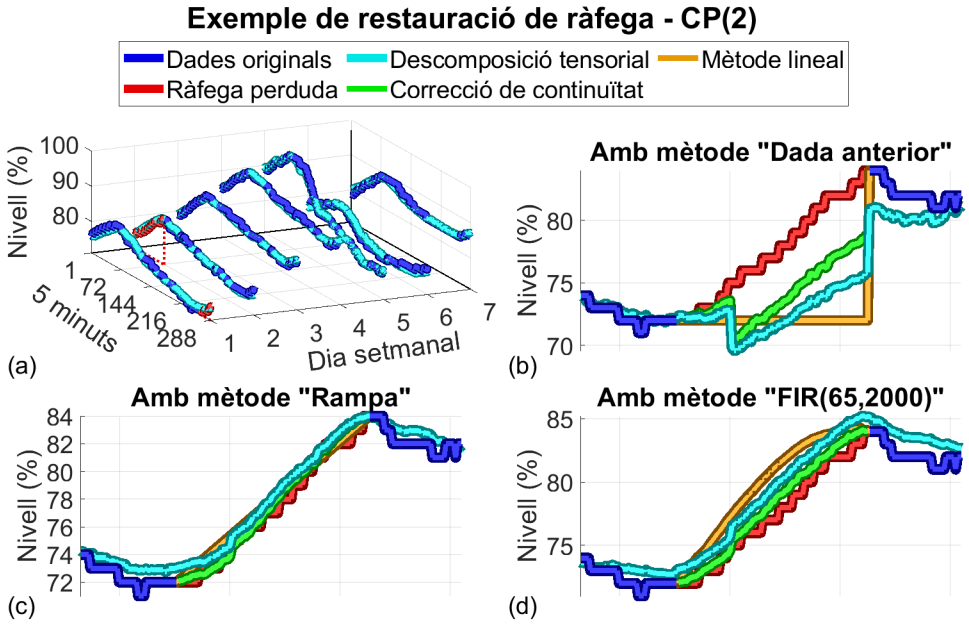

Figura 5.42: a) gràfic 3D del senyal original i de la descomposició CP(2) durant la setmana on es troba la ràfega. b), c, i d) resultat del tensor i de la correcció de la continuïtat per a cada un dels mètodes lineals descrits.

5.5. Conclusions

En aquest capítol es proposa una metodologia de restauració de dades que es divideix en dues etapes. En la primera s'utilitza un mètode lineal per omplir les dades perdudes, com per exemple algun dels mètodes lineals del capítol 4, ja que al fer servir tècniques de "maximization of expectations" és necessari omplir les dades buides. En la segona etapa les dades s'introdueixen en un tensor de tres dimensions ordenat de formar horària, diària i setmanal. És a dir, a la primera dimensió s'ordenen les dades segons la hora del dia (amb la resolució de 5 minuts de l'SCADA això suposen 288 mostres diàries). Un cop el tensor està ple amb les dades s'aplica un procés de simplificació, la descomposició, amb els models Tucker o CP. D'aquesta forma al recompondre el tensor la informació recuperada ignora algunes de les inconsistències generades per l'error d'estimació del mètode lineal de la primera etapa, capturant més fidelment les fluctuacions del senyal original. Després es realitza un procés senzill però important que aprofita aquestes fluctuacions per reconstruir les dades perdudes mantenint la continuïtat del senyal.

Segons els resultats obtinguts en aquest capítol es fa palès que el mètode tensorial proposat millora els resultats dels mètodes lineals, sobretot a mesura que la ràfega és fa més llarga com es mostra a la simulació de l'apartat *Mida de la ràfega*. A la figura 5.22 s'observa que fins a unes 150 mostres de ràfega, el mètode "Rampa" usat pels operaris i el mètode "FIR" proposat la secció 4.3, obtenen resultats semblants, a partir d 'aquest punt, el mètode "FIR" resulta una mica millor. Pel que fa al ús de tensors, aproximadament a partir de 50 o 100 mostres ja sembla que comencen a millorar els resultats respecte als mètodes lineals "Rampa" i "FIR". A l'apartat *Mida de la ràfega* es mostra una millora dels resultat al augmentar progressivament el nombre de setmanes de tensor n_w, fins arribar aproximadament a 7 setmanes, on sembla que s'estabilitzen es resultats. La metodologia proposada està dissenyada concretament per tractar dades perdudes de forma consecutiva, obtenint millors resultats, fins i tot, que l'algoritme CP-Wopt. Finalment, destacar que en la gran majoria dels casos, figures 5.24-5.42, el pas de la correcció de continuïtat de l'apartat *Correcció de la continuïtat* juga un paper molt important. De fet, és útil fins al punt de millorar els resultats de CP-Wopt, quan es combinen aquest algoritme ja conegut amb el procés proposat per la correcció de la continuïtat com es mostra a les figures 5.14-5.18.

Un detall més a favor de la metodologia, és que el temps d'excussió no és excessiu. Amb un Intel(R) Core(TM) i5-6200U de 2.3 GHz i 8GB de RAM amb Windows 7 Professional i utilitzant el Matlab 2018, com s'havia comentat anteriorment, el mètodes dels operaris que "Dada Anterior" i "Rampa" tarden només 0,2 i 4 ms respectivament. El mètode "FIR" demana de 2,5 a 10 segons, en canvi el procés de descomposició tensorial requereix només entre 0,1 i 0,5 segons.

Capítol 6

MÈTODE DE DOBLE DESCOMPOSICIÓ TENSORIAL

L'objectiu d'aquest capítol és refinar el mètode descrit al capítol anterior.

Per fer les simulacions s'utilitzen els tensors $\chi^{288\times7\times3}$ i $\chi^{288\times7\times7}$. Per extreure resultats es proven dos valors diferents per a la mida de les ràfegues: 100 i 200. Com que el mètode de reconstrucció amb tensors requereix un mètode lineal inicial, per fer les simulacions de les millores es fa servir el mètode "Rampa", degut a que el seu cost computacional a l'hora de realitzar les simulacions és molt menor que el del mètode "FIR".

A les seccions 6.1, 6.2 i 6.3 es descriuen cada una de les millores que s'han trobat i que ajuden a optimitzar la metodologia de reconstrucció de ràfegues de mostres perdudes mitjançant tensors proposada a la secció 5.2.

Finalment es recullen els resultats i les conclusions del capítol a les seccions 6.4 i 6.5.

6.1. Tensor centrat en ràfega

Per construir el tensor cal definir les seves dimensions per tal d'omplir-lo amb les dades corresponents. Com s'ha comentat el tensor configurat és $\chi^{288\times7\times n_w}$, on n_w indicaria la mida del sensor en nombre de setmanes de dades. En la metodologia inicial s'agafen les dades corresponents a la setmana on s'ha produït la ràfega i com a mínim una setmana abans i una setmana després, de forma que el tensor mínim és de 3 setmanes, $\chi^{288\times7\times3}$. Per construir tensors més grans s'agafa un nombre de setmanes més gran, abans i després de la setmana on es localitza la ràfega. Per aquest motiu en els experiments realitzats s'han fet servir tensors de mida senar, $n_w = 3, 5, 7,$ etc.

Al omplir el tensor d'aquesta forma, la ràfega queda situada a la setmana central, però no queda situada exactament al cor del tensor, ja que si la ràfega no està situada el dijous, no està al centre de la setmana, exactament. A més, la ràfega pot estar situada entre dos dies diferents, segons a quina hora comenci. A la figura 6.1 (a), (b), (c) i (d) es pot observar que no hi ha el mateix número de mostres davant que darrere de la ràfega perduda, sobretot si la ràfega comença

el dilluns o el diumenge. Hem observat que tot això provoca discontinuïtats en les estimacions obtingudes al reconstruir el senyal.

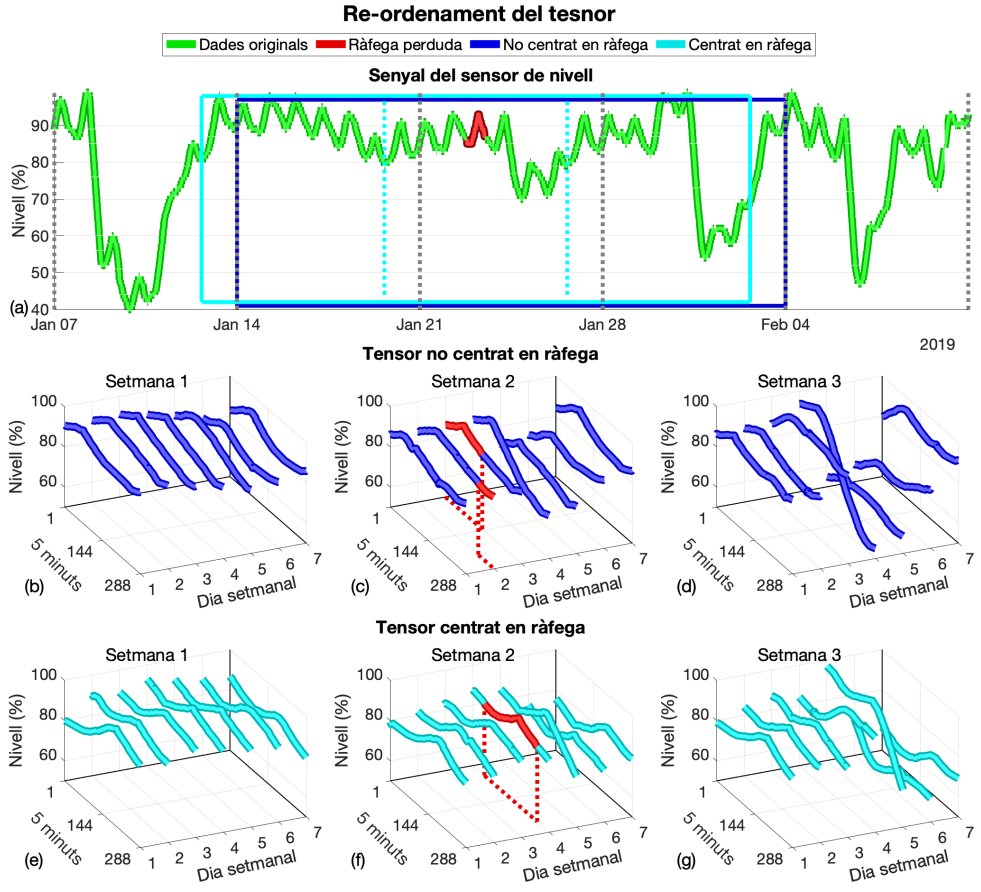

Figura 6.1: Senyal del sensor de nivell del dipòsit de Castell d'en Planes. Dos mètodes d'introducció de dades en un tensor $\chi^{288\times7\times3}$. Es mostra el posicionament d'una ràfega de 200 mostres amb cada un dels mètodes. (a) En verd trobem el senyal original i en vermell les dades eliminades. El requadre blau indica les dades agafades amb el mètode inicial o el verd el mètode millorat. (b), (c) i (d) mostren tres setmanes del mètode inicial que situa la ràfega a la setmana central, però no exactament al centre del tensor. (e), (f) i (g) mostren tres setmanes del mètode millorat que situa la ràfega justa al centre del tensor, al mig del dia i al mig de la setmana central del tensor.

Després d'identificar les causes que fan que en alguns casos les estimacions de les dades no siguin tant bones, hem procedit a millorar l'algorisme. Per tant, es realitza una modificació a l'algoritme que omple el tensor, forçant que la ràfega de dades perdudes quedi situada exactament al cor del tensor, és a dir, al mig del dia, al mig de la setmana i a la setmana del mig del tensor, figura 6.1 (a), (e), (f) i (g). Amb l'ordenament inicial del tensor, la ràfega només

es situava exactament al cor del tensor quan aquesta es produïa el dijous (el dia central de la setmana si comencem a comptar des de dilluns). I per ser precisos, hauria de ser just al mig del dijous, ja que tenint en compte que es limiten les ràfegues a menys d'un dia de durada, això sempre seria possible. La nova manera proposada d'omplir el tensor força que la ràfega es situï en aquesta posició. Llavors, si B és el número de mostres de la ràfega i $\chi^{I\times J\times K}$ el tensor, la ràfega es situa a $J=4$, $K=0.5(n_w+1)$, col·locant la primera mostra a I_i=0.5(288-B). Per tant, la ràfega ocuparia les posicions $\chi^{I_i:I_i+B-1\times 4\times 0.5(n_w+1)}$ dins el tensor. Degut a aquesta nova manera d'introduir les dades al tensor, i vist des del punt de vista del nou tensor, els dies pràcticament mai comencen a les 00:00 i les setmanes poques vegades comencen el dilluns. A canvi sempre hi ha el mateix número de dades abans i després de la ràfega, cosa que quasi mai passava amb l'anterior manera d'omplir el tensor.

6.2. Suavitzat del senyal

El senyal original del nivell de dipòsit té un aspecte escalonat, com d'un senyal digitalitzat, degut a les condicions de consum pel que fa a omplir i buidar el dipòsit i a la resolució del sensor. No obstant, el procés tensorial al que es sotmeten les dades per realitzar la reconstrucció, igual que passa amb els mètodes lineals, entrega una resposta analògica, és a dir, el senyal recuperat no té l'aspecte esgraonat. Degut a això, es va considerar l'opció de dissenyar un algoritme de suavitzat del senyal, per aplicar-lo abans d'introduir el senyal al tensor, de forma que el senyal d'entrada ja no tingui l'aspecte escalonat.

L'algoritme està dissenyat específicament per la manera de treballar del sensor de nivell estudiat. Hi ha mètodes més elaborats, com ara els que utilitzen tècniques de filtratge, però que poden introduir retards degut al retard de grup dels filtres o poden dificultar la recuperació del senyal original en el cas de d'estar-hi interessats. Amb aquest mètode es recupera el senyal senzillament arrodonint a l'enter més proper. Les mostres es processen en grups de dades consecutives del mateix valor enter i tenint en compte si el senyal està creixent, decreixent o si està en un mínim o màxim relatiu. Per tant, aquests blocs de dades consecutives amb un mateix valor, que anomenem A, es processen tenint en compte els valors dels blocs contigus. Si el bloc correspon a un moment on el senyal està creixent, es dibuixa una línia recta de pendent positiu entre $A-0.49$ i $A+0.49$. Si està decreixent es dibuixa una línia recta de pendent

negatiu entre $A + 0.49$ i $A - 0.49$. Si en canvi es tracta d'un mínim o màxim relatiu, es dibuixa una forma triangular amb la corresponent orientació. A les figures 6.2-6.5 se'n mostren uns exemples. Amb aquest procediment s'obté una millora dels resultats força estable en totes les condicions simulades, respecte a la mida del tensor i de la ràfega.

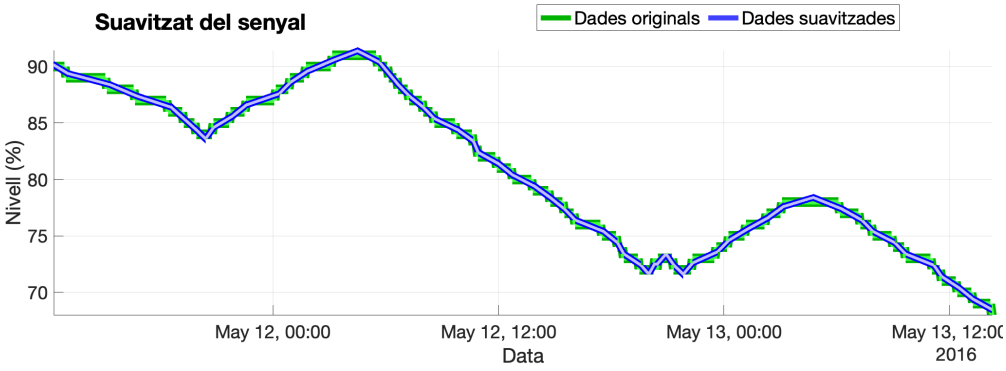

Figura 6.2: Procés de suavitzat del senyal aplicat abans de la descomposició.

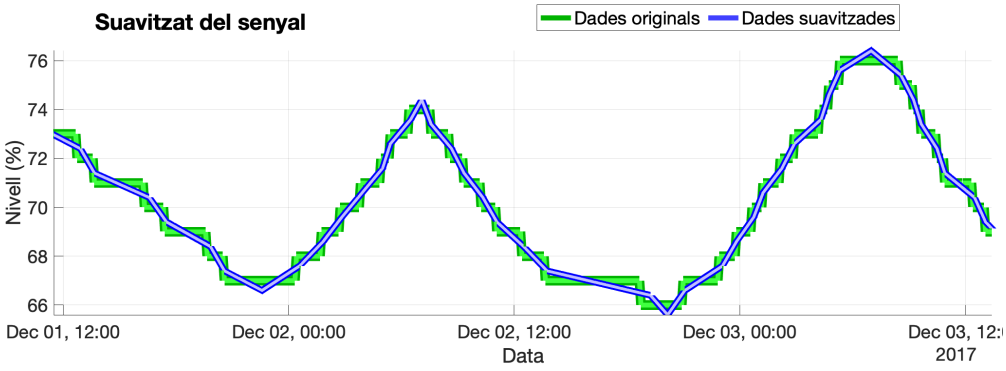

Figura 6.3: Procés de suavitzat del senyal aplicat abans de la descomposició.

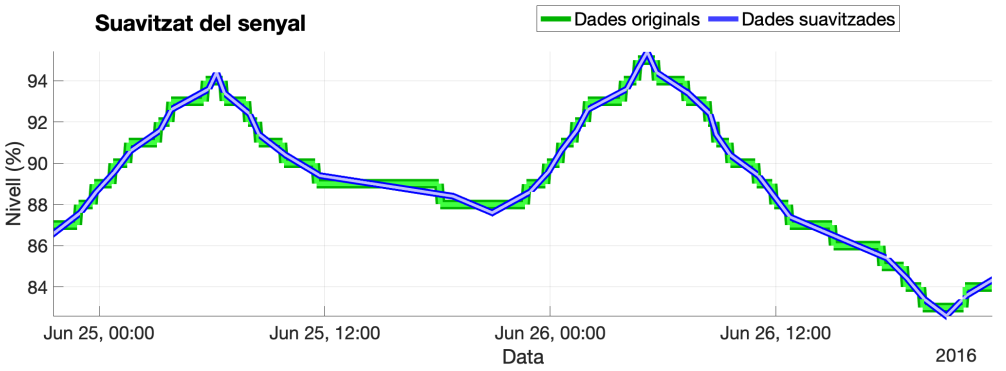

Figura 6.4: Procés de suavitzat del senyal aplicat abans de la descomposició.

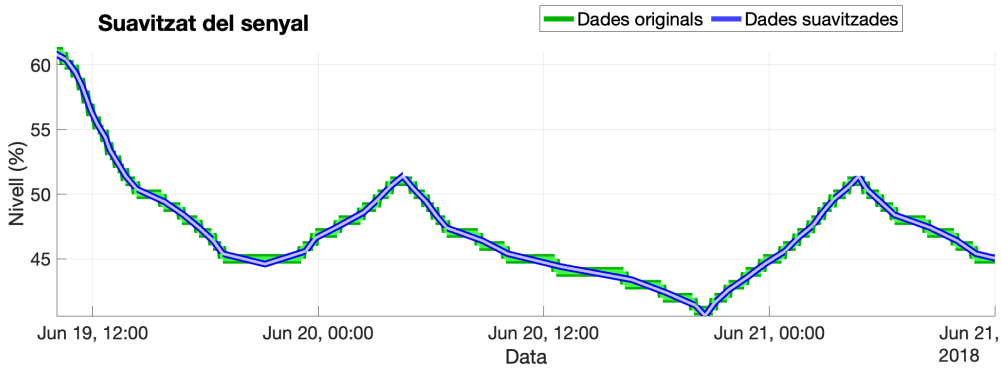

Figura 6.5: Procés de suavitzat del senyal aplicat abans de la descomposició.

6.3. Doble descomposició tensorial

Aquesta millora es centra en la configuració de la descomposició tensorial. Al realitzar les diferents simulacions, s'aprecia que segons el mètode lineal utilitzat per omplir les dades perdudes abans d'introduir-les al tensor, la configuració òptima del nucli tensorial que s'utilitza en la descomposició pot ser diferent (tant en el cas del model Tucker com del model CP). En concret, com menys eficaç és el mètode lineal (en termes de l'MSE calculat) més simple o petit ha de ser el nucli a utilitzar. És a dir, més petites han de ser les dimensions del nucli de la descomposició.

Tenint en compte això, es planteja un mètode que utilitza dos cops el procediment de descomposició tensorial amb dos nuclis diferents. El primer nucli es dissenya petit i simple, per tal d'obtenir una bona aproximació encara que el mètode lineal usat per fer la primera estimació de valors provoqui un error gran. En la segona descomposició es fa servir un nucli més gran, ja que l'estimació inicial és més bona, gràcies a haver estat refinada per la primera descomposició tensorial.

Per ajudar a clarificar l'esquema proposat, a la figura 6.6, es mostra un diagrama per cada un dels models de descomposició que s'han tractat en la tesi, el Tucker i el CP. El senyal original amb la ràfega de valors perduts és $\boldsymbol{x}$. En el primer pas, després del suavitzat, s'aplica un mètode lineal que realitza una primera estimació d'aquests valors $\boldsymbol{x}$. S'utilitza el mètode "Rampa" perquè a més de ser senzill és relativament eficaç. A continuació s'aplica la primera descomposició tensorial a $\chi^{288\times 7\times n_w}$ amb l'objectiu d'obtenir una aproximació més bona que la obtinguda amb el mètode lineal i mitjançant un nucli petit. En l'exemple a) de la figura 6.6, el model Tucker, es fa servir la configuració

Tucker (4,6,1). Com a resultat s'obté $\chi_{(1)}^{288\times 7\times n_w}$, on s'aplica el pas de la correcció de continuïtat a les mostres situades a la posició de la ràfega perduda, per tal d'aprofitar les fluctuacions capturades en la primera descomposició i millorar l'estimació feta amb el mètode lineal. El tensor $\chi'^{288\times 7\times n_w}_{(1)}$ s'omple amb les dades originals substituint les posicions buides de la ràfega perduda pels valors adaptats de $\chi_{(1)}^{288\times 7\times n_w}$ amb la correcció de la continuïtat. El següent pas consisteix en aplicar una segona descomposició, aquest cop amb un nucli més gran. En l'exemple a) del model Tucker es fa sevrir la configuració Tucker(4,7,7)). Com a resultat s'obté $\chi_{(2)}^{288\times 7\times n_w}$, que és una aproximació de $\chi'^{288\times 7\times n_w}_{(1)}$. De nou, a les posicions de la ràfega perduda de $\chi_{(2)}^{288\times 7\times n_w}$ s'aplica el procés de correcció de continuïtat per obtenir el resultat final. En l'exemple, que es mostra a continuació, s'apliquen les configuracions Tucker(4,6,1) i Tucker(4,7,7) ja que resulten ser les òptimes en l'estudi realitzat a l'apartat *Configuració òptima* per a ràfegues de 200 mostres i un tensor $\chi^{288\times 7\times 7}$. Els valors òptims en cas del model CP, són CP(1) i CP(15) per a les mateixes mides de tensor i ràfega.

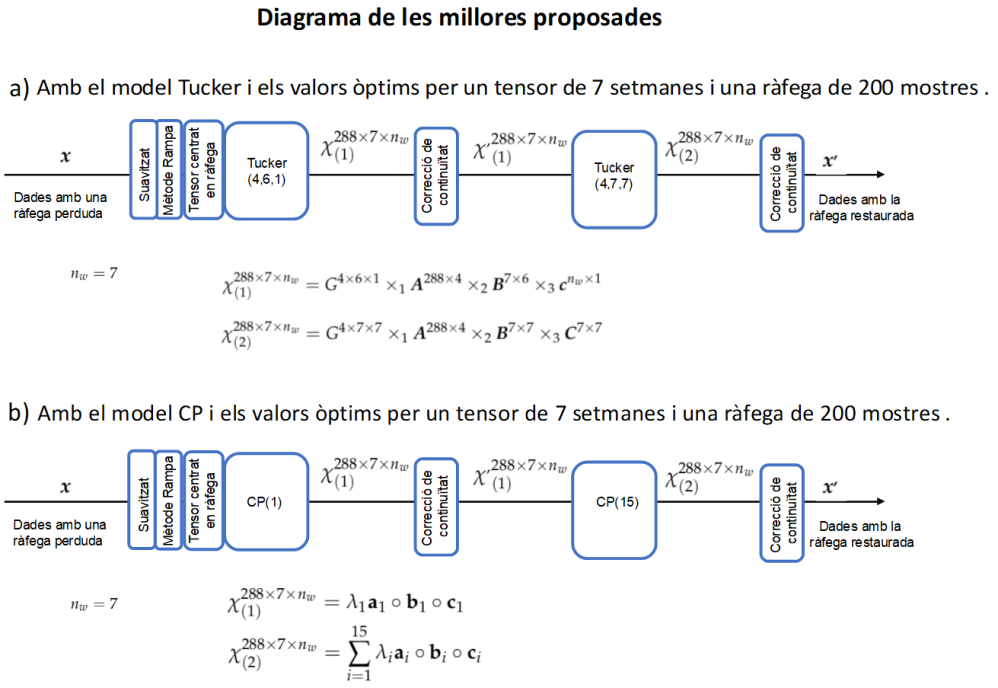

Figura 6.6: Diagrama de la doble descomposició proposada per als dos models, Tucker (a) i CP (b). En l'exemple es mostren els valors òptims per al cas de $n_w = 7$.

6.3.1. Configuració òptima

Per determinar el millor nucli posssible de la primera descomposició es realitza un experiment que consisteix en fer un conjunt de simulacions aplicant només una descomposició i fent servir molts nuclis diferents. Calculant l'MSE per mostra amb els diferents nuclis provats, es podrà seleccionar el millor nucli per la primera descomposició. Aquest experiment es realitza per les diferents condicions de mida de tensor (3 i 7 setmanes) i de ràfega (100 i 200 mostres).

Per determinar el millor nucli possible de la segona descomposició es fixa el nucli de la primera descomposició d'acord als valors òptims determinats en el primer experiment. Això es fa per cada mida de tensor (3/7) i per cada longitud de ràfega (100/200). A continuació es realitza el mateix experiment, calculant l'MSE, i variant la mida dels nuclis a la segona descomposició. Aquesta exploració amb diferents condicions serveix per veure que el mètode és molt robust ja que les configuracions òptimes en les dues etapes de la descomposició tensorial, per tots els casos de mida de tensor i de ràfega, és similar.

A les figures 6.7-6.10 es mostren els resultats d'aquest experiment en el cas del model CP. Es pot observar que amb totes les condicions s'obtenen resultats similars. En aquests casos, la sub-figura (a) mostra l'experiment amb una sola descomposició tensorial, per determinar el nucli òptim de la primera descomposició. La sub-figura (b) mostra el segon experiment per determinar el millor nucli per la segona descomposició on s'ha fixat la mida del primer nucli amb la mida obtinguda al primer experiment. En vermell s'indica el cas de menor MSE i en verd els casos que no superen en un 5% l'error mínim. Resulta interessant comprovar que en totes les condicions provades el millor nucli per a la primera descomposició és el més simple, el CP(1). Per a la segona descomposició, tot i que el mínim no és sempre el mateix, seleccionar la configuració més alta possible és la millor opció (en l'exemple la CP(15)), ja que a partir d'un cert punt s'estabilitza l'error i per tant, encara que en un cas concret hi hagi una configuració amb un error una mica menor a l'obtingut amb la CP(15), la diferencia entre els errors és extremadament petita. En tot cas, la selecció del nucli de la primera descomposició és molt robusta i ha de ser el CP(1). El de la segona ha de ser més gran i es disposa d'un ampli ventall d'opcions bones (configuracions indicades amb color verd a les figures) ja que un cop s'obté el mínim possible, seleccionar una nucli més gran produeix resultats molt semblants.

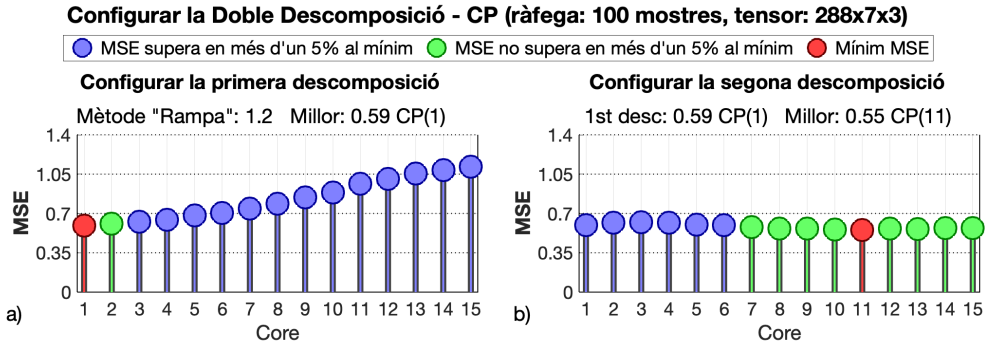

Figura 6.7: Configuració òptima de la Doble descomposició amb model CP, tensor $\chi^{288\times7\times3}$ i ràfega de 100. (a) MSE de la primera descomposició per diferents nuclis. (b) MSE de la segona descomposició usant el nucli òptim per la primera $G^{1\times1\times1}$, i varis nuclis per la segona.

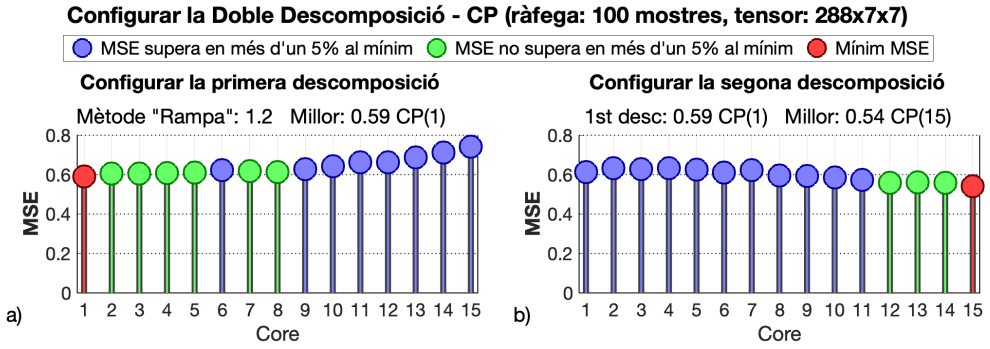

Figura 6.8: Configuració òptima de la Doble descomposició amb model CP, tensor $\chi^{288\times7\times7}$ i ràfega de 100. (a) MSE de la primera descomposició per diferents nuclis. (b) MSE de la segona descomposició usant el nucli òptim per la primera $G^{1\times1\times1}$, i varis nuclis per la segona.

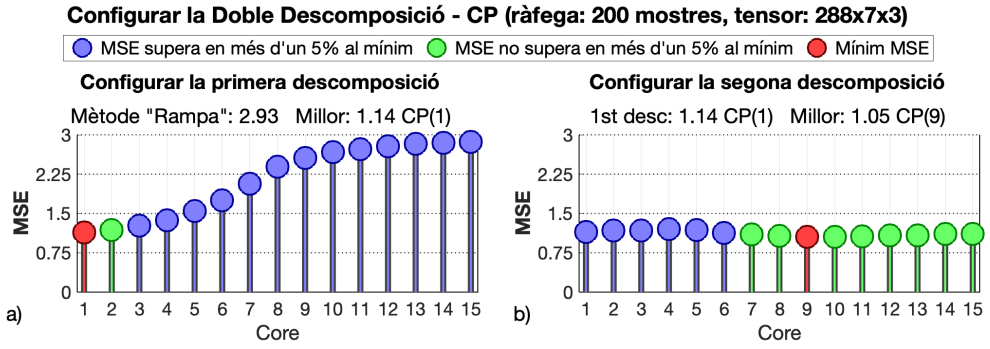

Figura 6.9: Configuració òptima de la Doble descomposició amb model CP, tensor $\chi^{288\times7\times3}$ i ràfega de 200. (a) MSE de la primera descomposició per diferents nuclis. (b) MSE de la segona descomposició usant el nucli òptim per la primera $G^{1\times1\times1}$, i varis nuclis per la segona.

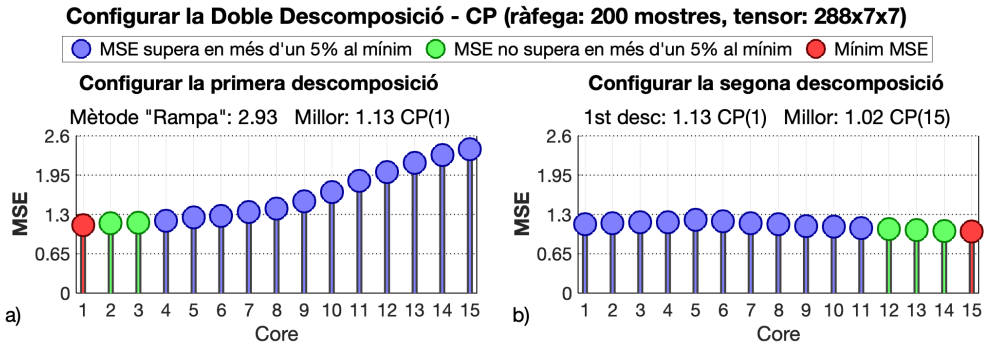

Figura 6.10: Configuració òptima de la Doble descomposició amb model CP, tensor $\chi^{288\times7\times7}$ i ràfega de 200. (a) MSE de la primera descomposició per diferents nuclis. (b) MSE de la segona descomposició usant el nucli òptim per la primera $G^{1\times1\times1}$, i varis nuclis per la segona.

A les figures 6.11-6.34 es mostren els resultats d'aquest experiment en el cas del model Tucker. Aquest cas es més costós en termes de capacitat de càlcul i de visualització, ja que en aquest model hi ha molts més paràmetres que en el model CP per configurar el nucli del tensor. En el model Tucker el número de paràmetres depèn de la dimensió del tensor, en el cas del tensor de tres dimensions són tres paràmetres. Per poder observar els resultats gràficament, al haver-hi tres paràmetres, cal fixar-ne un. Es fixa el que fa referència al nombre de setmanes del tensor, n_w. Llavors es mostra una representació visual de l'MSE obtingut amb diferents combinacions dels altres dos paràmetres. Per cada figura la primera columna representa l'experiment realitzat per determinar el millor nucli per la primera descomposició i la segona columna l'experiment per trobar el millor nucli de la segona descomposició un cop fixada la primera segons s'indiqui. En vermell s'indica el cas de menor MSE i en verd els casos que no superen en un 5 % l'error mínim.

En les figures es poden veure grups de punts verds, que representen nuclis amb resultats molt semblants a l'òptim. Això significa que hi ha un seguit de configuracions amb resultats molt semblants en termes de MSE. En general sembla que la millor opció és seleccionar el mínim valor possible en el paràmetre relatiu al número de setmanes del tensor, n_w, en la primera descomposició, i després seleccionar el número màxim en la segona. Els altres dos paràmetres semblen variar més, però en general el que fa referència al número de dies de la setmana ha de ser gran, prop del màxim 7). En canvi el que fa referència a les hores del dia ha de ser més petit, sobretot comparat amb el seu màxim valor possible, 288.

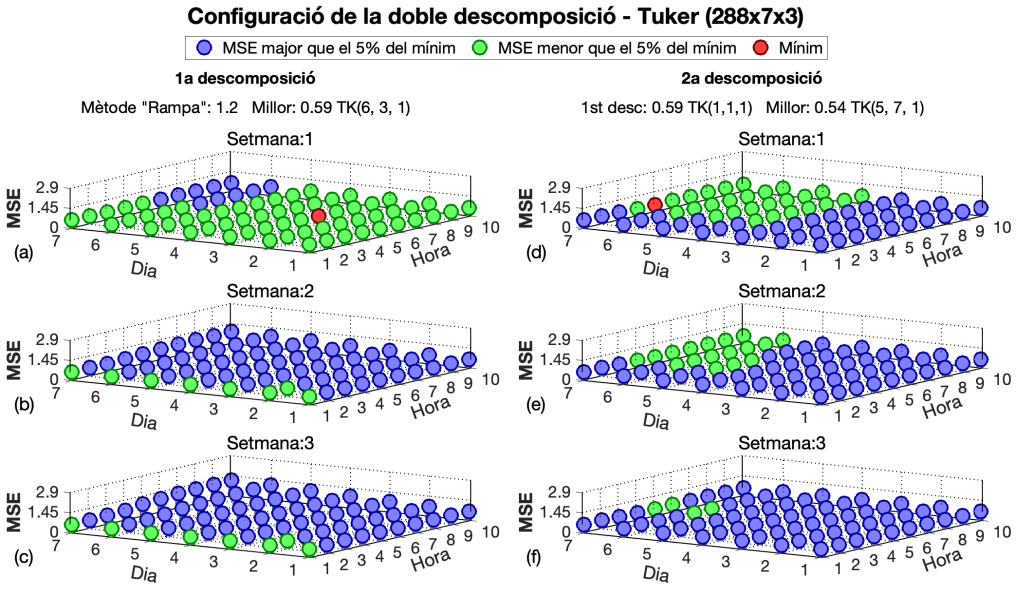

Figura 6.11: Configuració de la doble descomposició amb model Tucker, tensor $\chi^{288\times7\times3}$ i ràfega de 100. (a) MSE de la primera descomposició per diferents nuclis. (b) MSE de la segona descomposició per varis nuclis i usant el nucli $G^{1\times1\times1}$ a la primera (el nucli més simple possible).

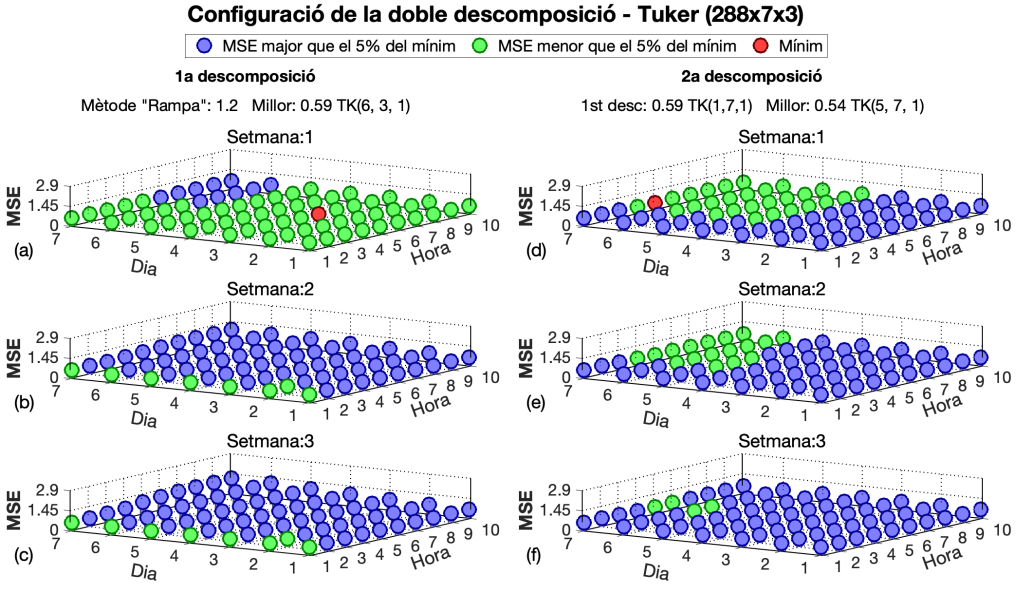

Figura 6.12: Configuració de la doble descomposició amb model Tucker, tensor $\chi^{288\times7\times3}$ i ràfega de 100. (a) MSE de la primera descomposició per diferents nuclis. (b) MSE de la segona descomposició per varis nuclis i usant el nucli $G^{1\times7\times1}$ a la primera (un nucli que simplifica hores i setmanes però manté la mida de la dimensió setmanal).

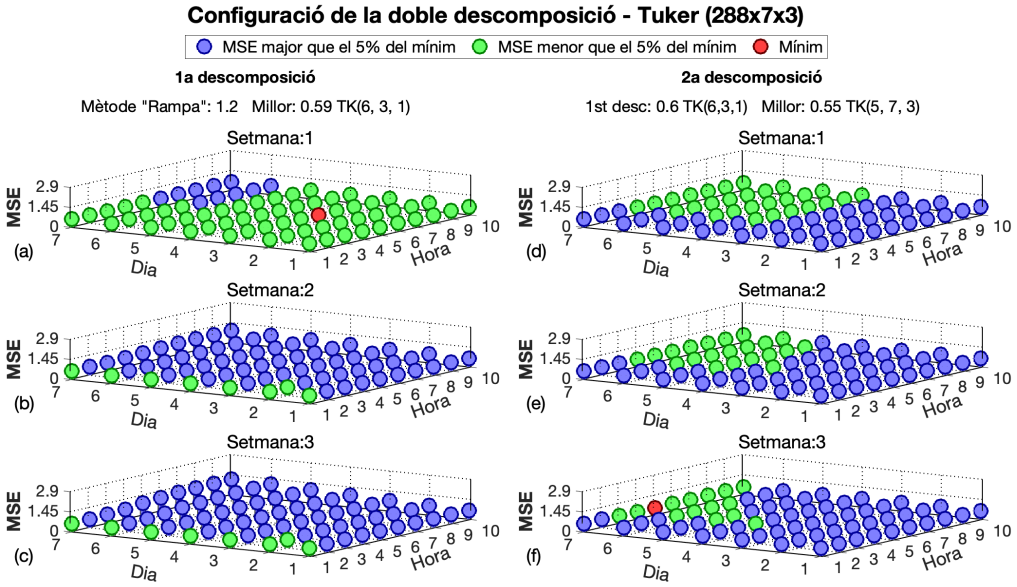

Figura 6.13: Configuració de la doble descomposició amb model Tucker, tensor $\chi^{288\times7\times3}$ i ràfega de 100. (a) MSE de la primera descomposició per diferents nuclis. (b) MSE de la segona descomposició per varis nuclis i usant el nucli $G^{6\times3\times1}$ a la primera (l'òptim en aquestes condicions de tensor i mida de ràfega).

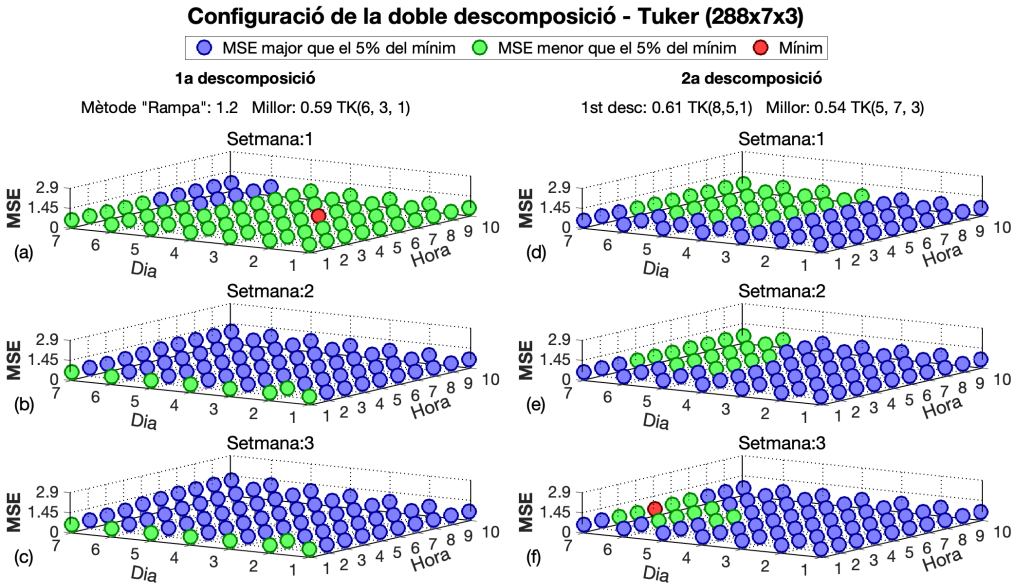

Figura 6.14: Configuració de la doble descomposició amb model Tucker, tensor $\chi^{288\times7\times3}$ i ràfega de 100. (a) MSE de la primera descomposició per diferents nuclis. (b) MSE de la segona descomposició per varis nuclis i usant el nucli $G^{8\times5\times1}$ a la primera (l'òptim en un tensor $\chi^{288\times7\times7}$ i 100 mostres de ràfega).

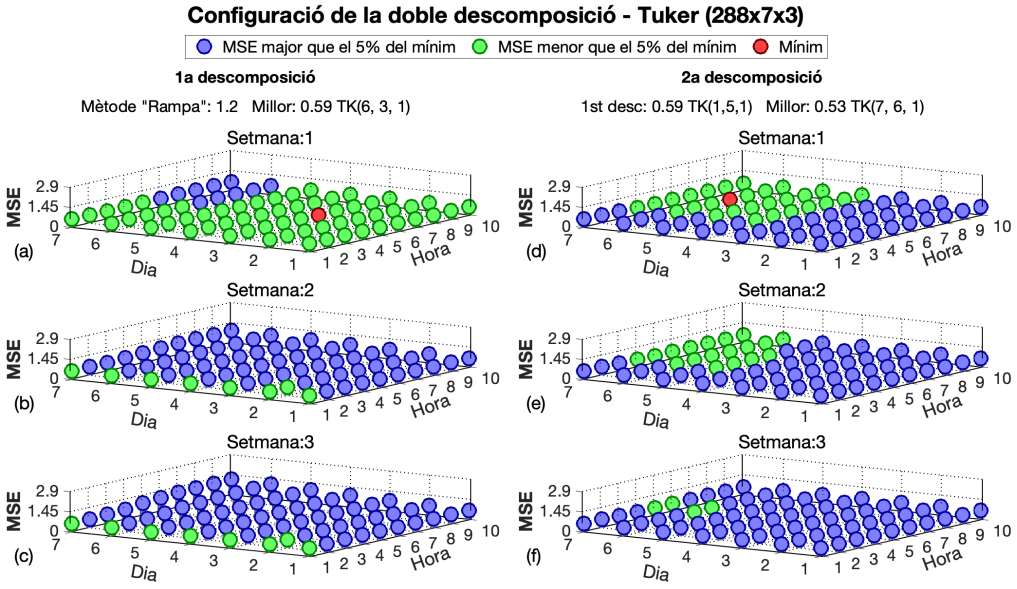

Figura 6.15: Configuració de la doble descomposició amb model Tucker, tensor $\chi^{288\times7\times3}$ i ràfega de 100. (a) MSE de la primera descomposició per diferents nuclis. (b) MSE de la segona descomposició per varis nuclis i usant el nucli $G^{1\times5\times1}$ a la primera (l'òptim en un tensor $\chi^{288\times7\times3}$ i 200 mostres de ràfega).

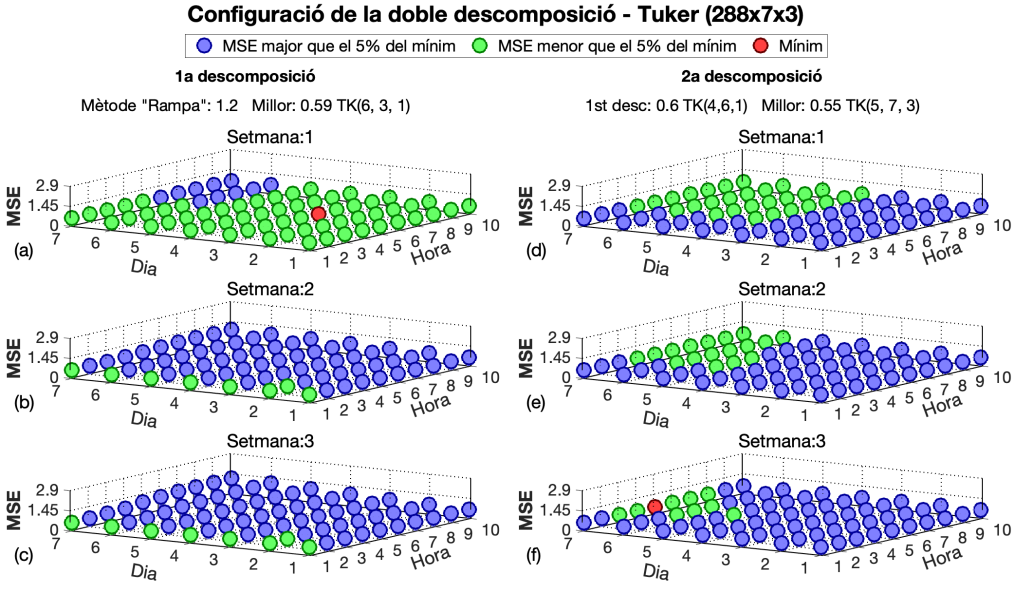

Figura 6.16: Configuració de la doble descomposició amb model Tucker, tensor $\chi^{288\times7\times3}$ i ràfega de 100. (a) MSE de la primera descomposició per diferents nuclis. (b) MSE de la segona descomposició per varis nuclis i usant el nucli $G^{4\times6\times1}$ a la primera (l'òptim en un tensor $\chi^{288\times7\times7}$ i 200 mostres de ràfega).

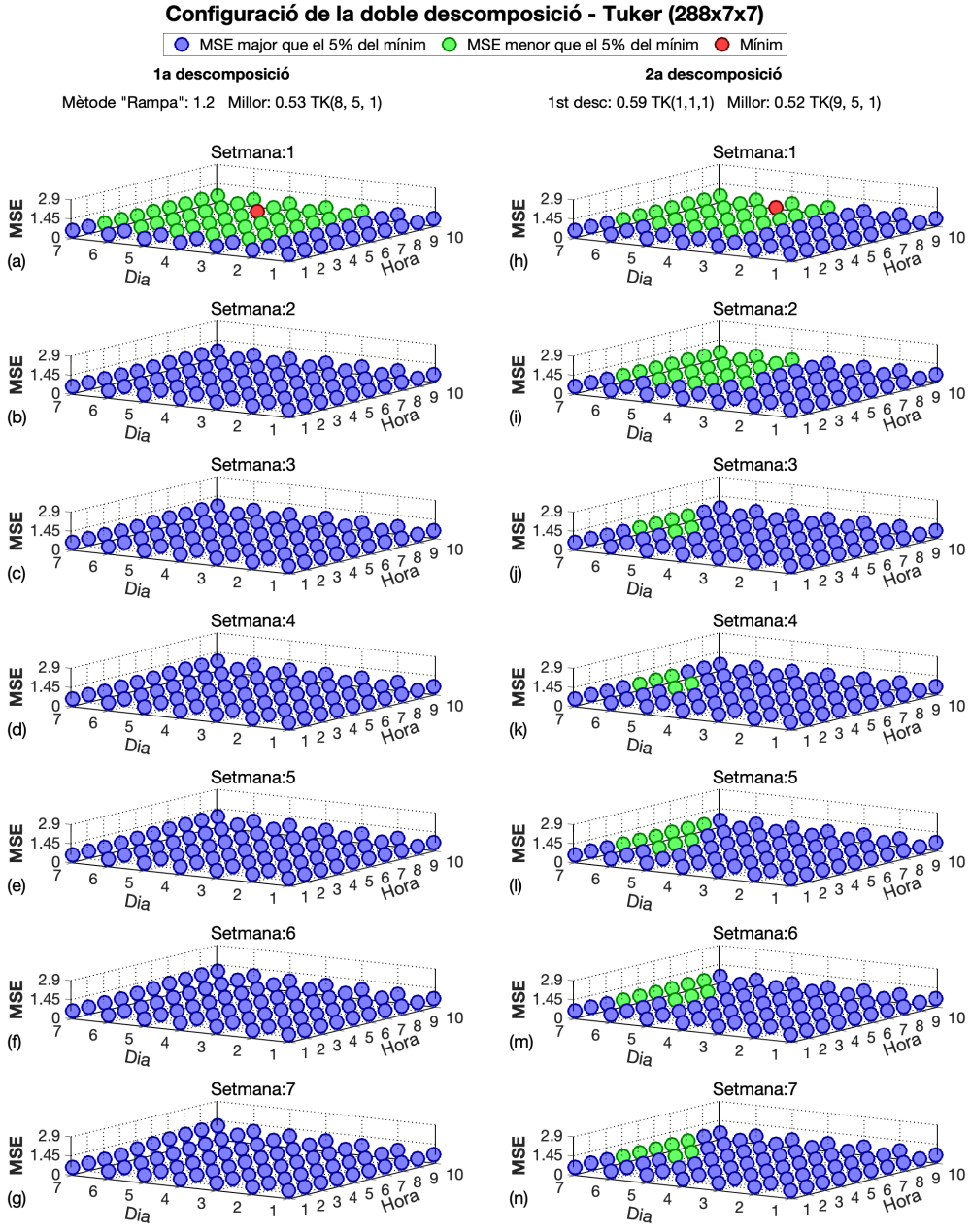

Figura 6.17: Configuració de la doble descomposició amb model Tucker, tensor $\chi^{288\times7\times7}$ i ràfega de 100. (a) MSE de la primera descomposició per diferents nuclis. (b) MSE de la segona descomposició per varis nuclis i usant el nucli $G^{1\times1\times1}$ a la primera (el nucli més simple possible).

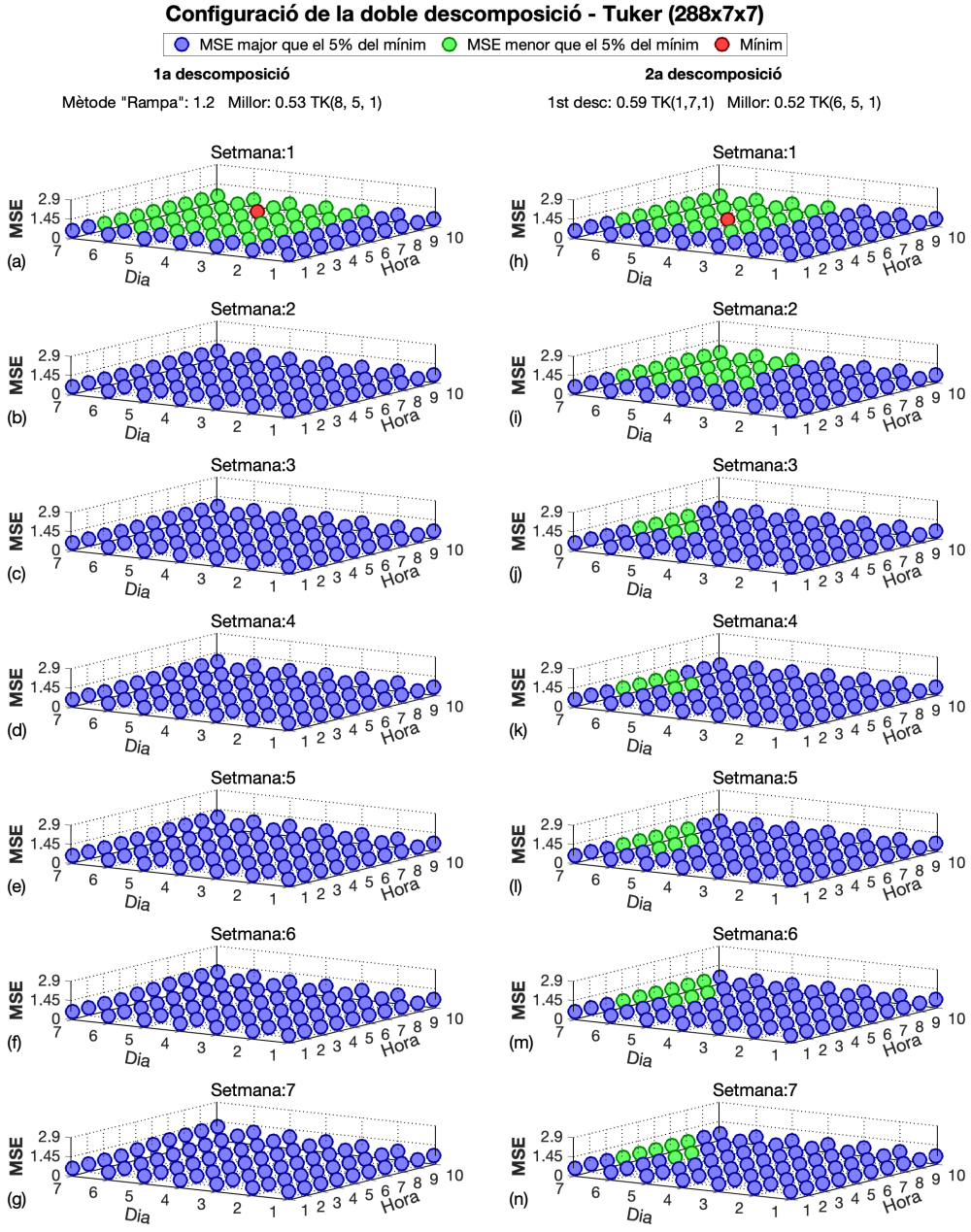

Figura 6.18: Configuració de la doble descomposició amb model Tucker, tensor $\chi^{288\times7\times7}$ i ràfega de 100. (a) MSE de la primera descomposició per diferents nuclis. (b) MSE de la segona descomposició per varis nuclis i usant el nucli $G^{1\times7\times1}$ a la primera (un nucli que simplifica hores i setmanes però manté la mida de la dimensió setmanal).

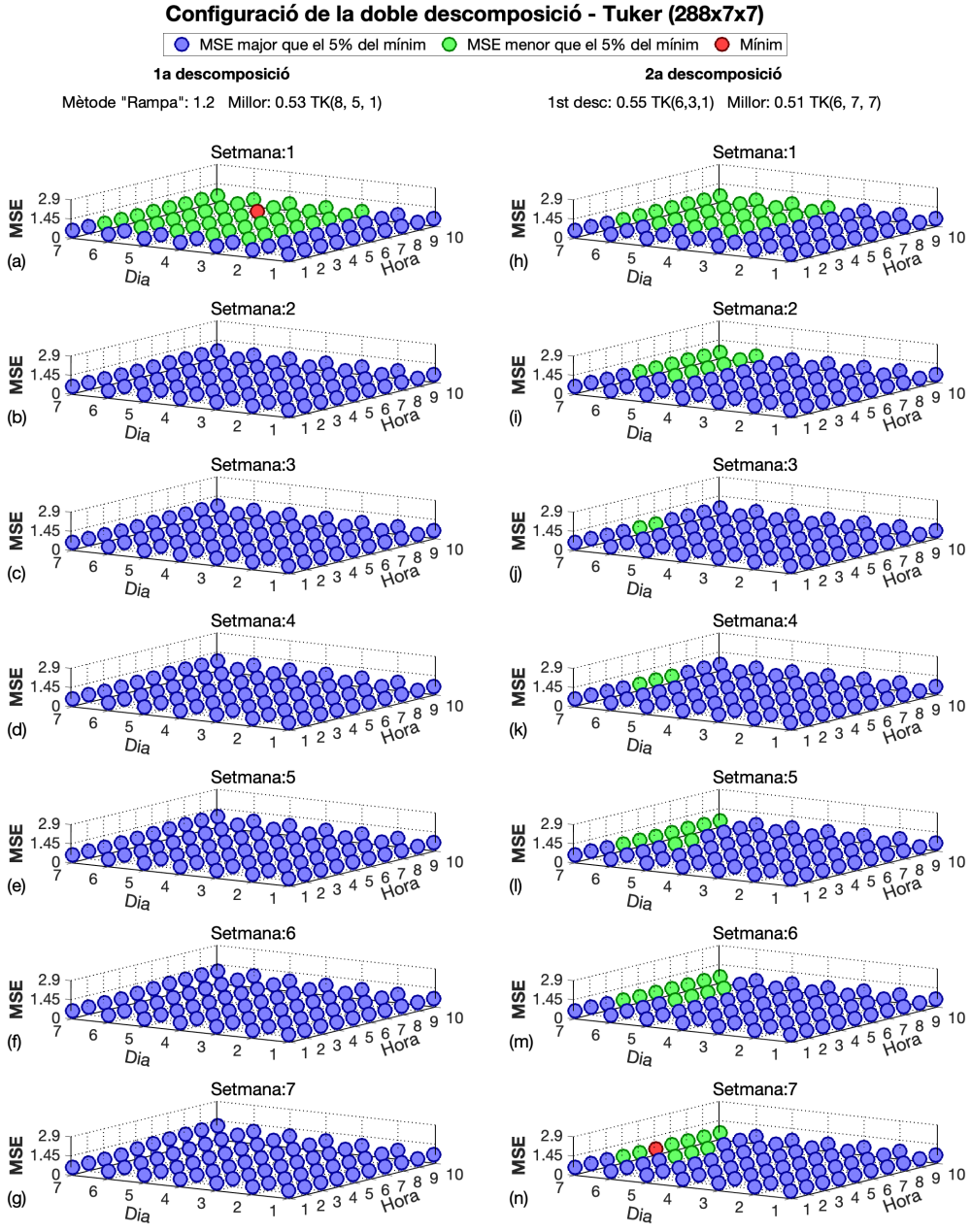

Figura 6.19: Configuració de la doble descomposició amb model Tucker, tensor $\chi^{288\times7\times7}$ i ràfega de 100. (a) MSE de la primera descomposició per diferents nuclis. (b) MSE de la segona descomposició per varis nuclis i usant el nucli $G^{6\times3\times1}$ a la primera (l'òptim en un tensor $\chi^{288\times7\times3}$ i 100 mostres de ràfega).

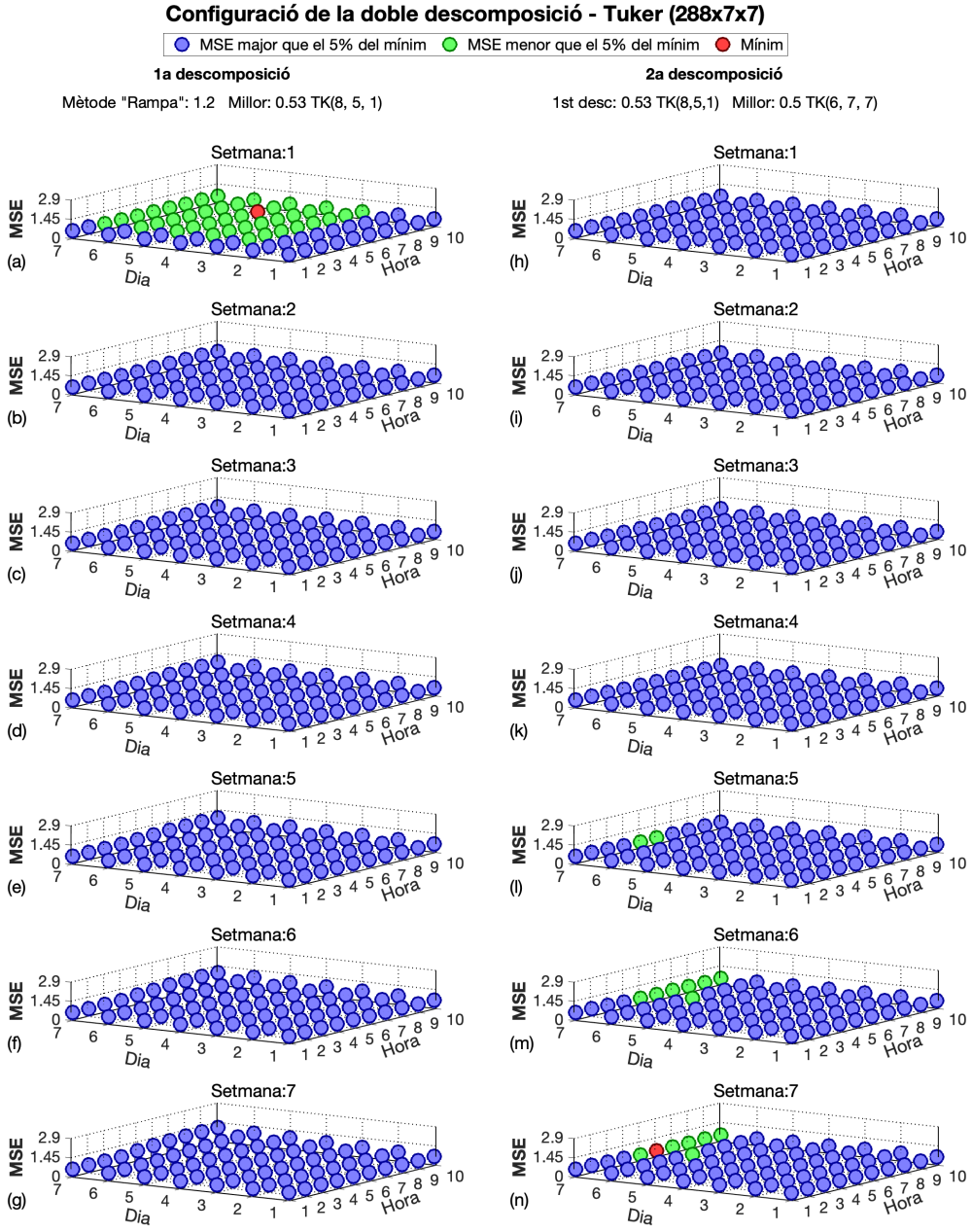

Figura 6.20: Configuració de la doble descomposició amb model Tucker, tensor $\chi^{288\times7\times7}$ i ràfega de 100. (a) MSE de la primera descomposició per diferents nuclis. (b) MSE de la segona descomposició per varis nuclis i usant el nucli $G^{8\times5\times1}$ a la primera (l'òptim en aquestes condicions de tensor i mida de ràfega).

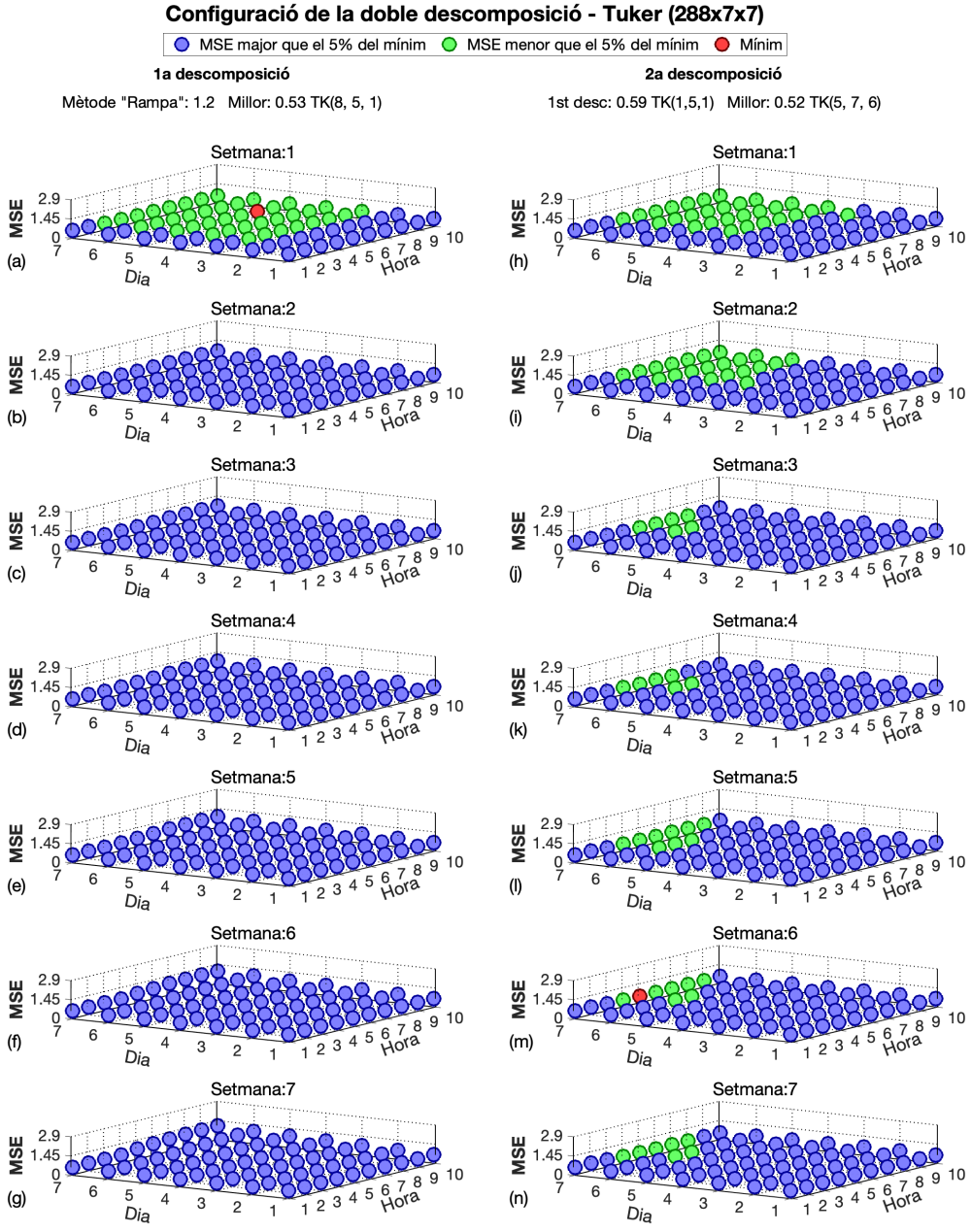

Figura 6.21: Configuració de la doble descomposició amb model Tucker, tensor $\chi^{288\times7\times7}$ i ràfega de 100. (a) MSE de la primera descomposició per diferents nuclis. (b) MSE de la segona descomposició per varis nuclis i usant el nucli $G^{1\times5\times1}$ a la primera (l'òptim en un tensor $\chi^{288\times7\times3}$ i 200 mostres de ràfega).

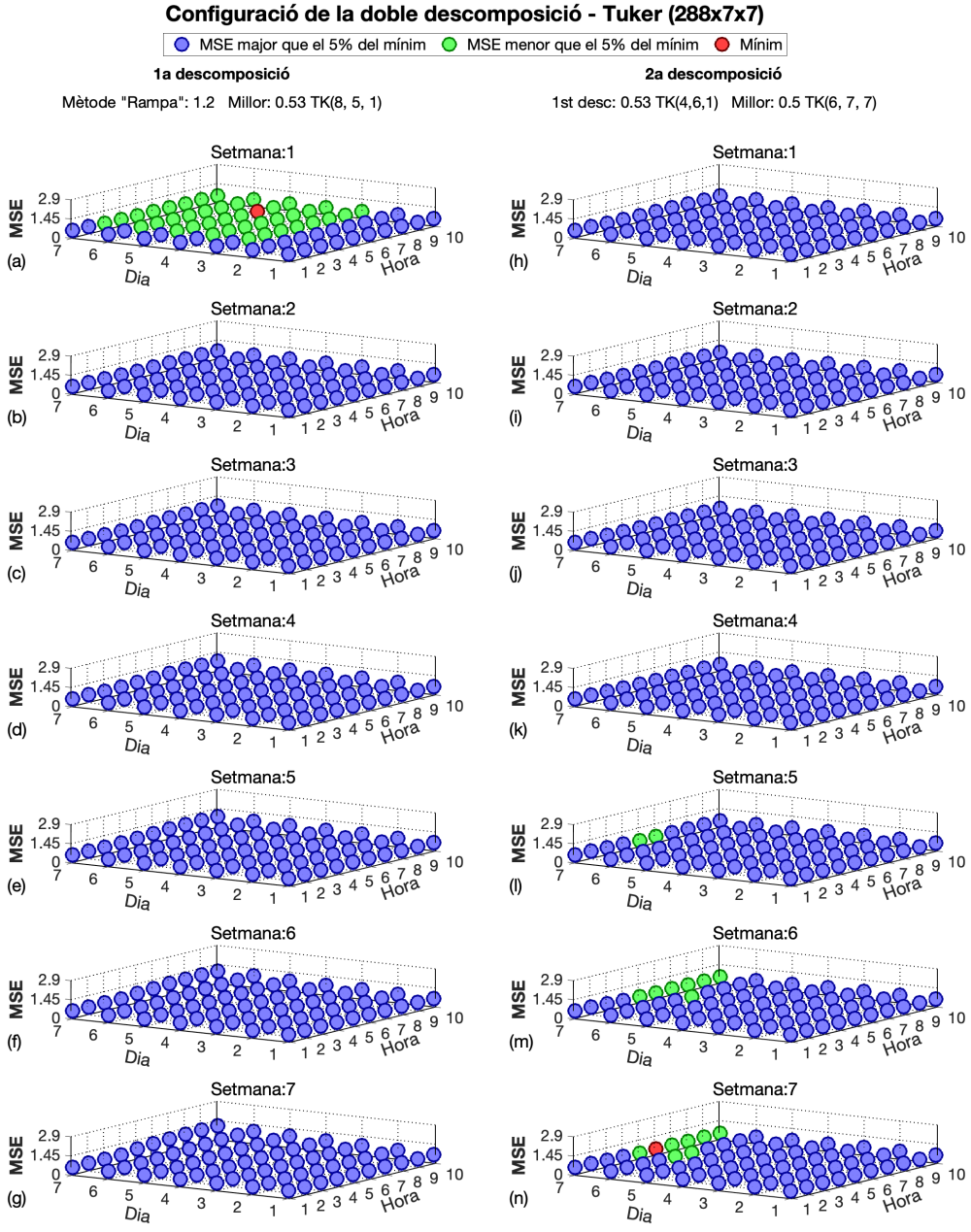

Figura 6.22: Configuració de la doble descomposició amb model Tucker, tensor $\chi^{288\times7\times7}$ i ràfega de 100. (a) MSE de la primera descomposició per diferents nuclis. (b) MSE de la segona descomposició per varis nuclis i usant el nucli $G^{4\times6\times1}$ a la primera (l'òptim en un tensor $\chi^{288\times7\times7}$ i 200 mostres de ràfega).

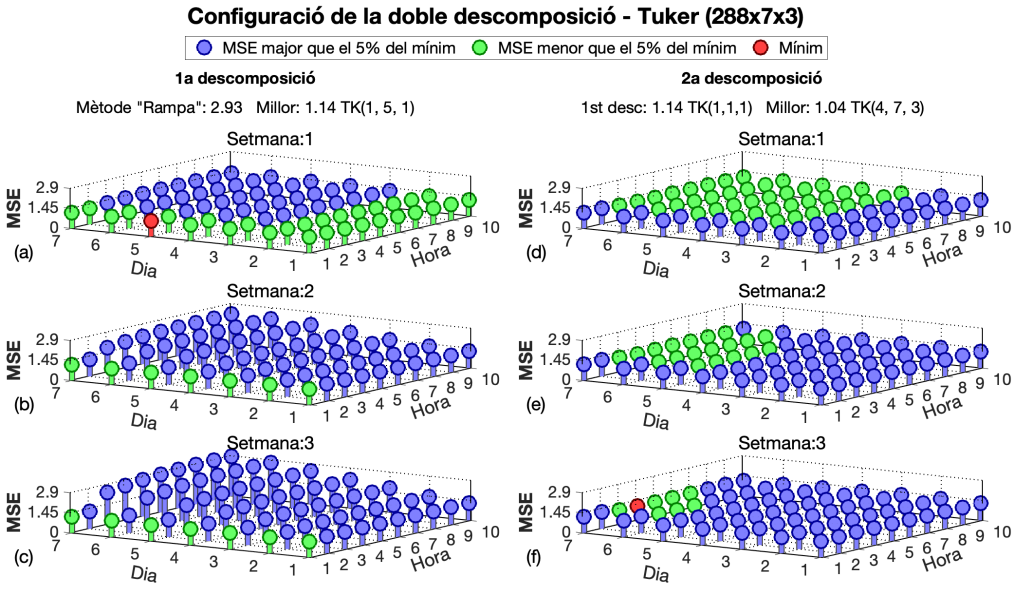

Figura 6.23: Configuració de la doble descomposició amb model Tucker, tensor $\chi^{288\times7\times3}$ i ràfega de 200. (a) MSE de la primera descomposició per diferents nuclis. (b) MSE de la segona descomposició per varis nuclis i usant el nucli $G^{1\times1\times1}$ a la primera (el nucli més simple possible).

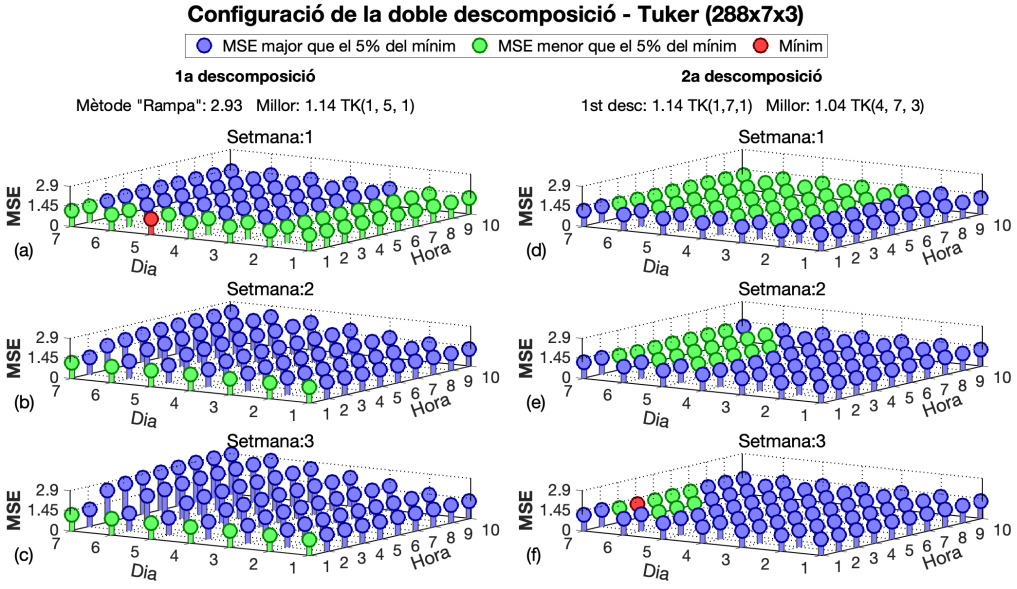

Figura 6.24: Configuració de la doble descomposició amb model Tucker, tensor $\chi^{288\times7\times3}$ i ràfega de 200. (a) MSE de la primera descomposició per diferents nuclis. (b) MSE de la segona descomposició per varis nuclis i usant el nucli $G^{1\times7\times1}$ a la primera (un nucli que simplifica hores i setmanes però manté la mida de la dimensió setmanal).

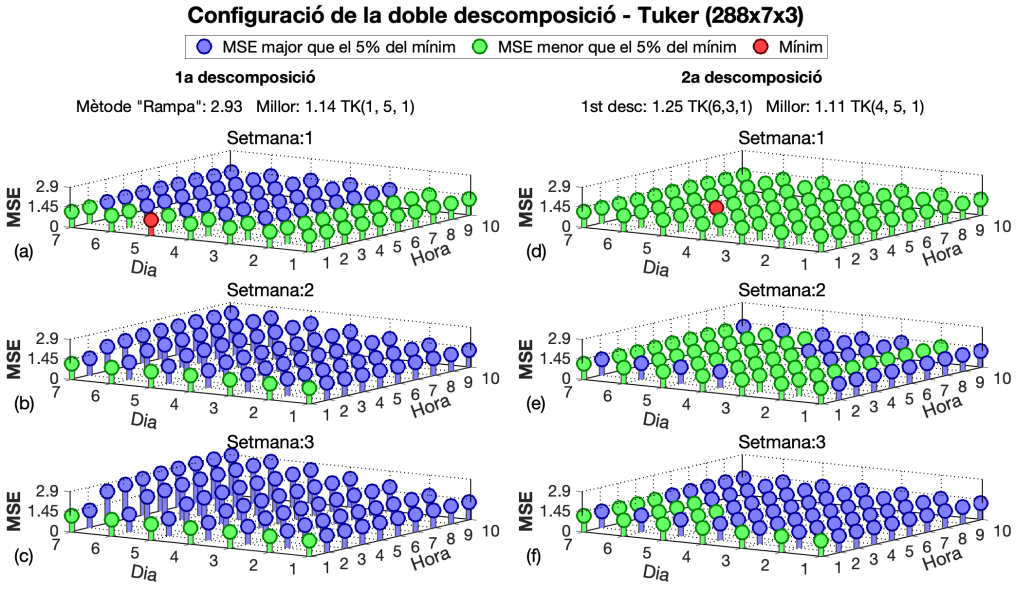

Figura 6.25: Configuració de la doble descomposició amb model Tucker, tensor $\chi^{288\times7\times3}$ i ràfega de 200. (a) MSE de la primera descomposició per diferents nuclis. (b) MSE de la segona descomposició per varis nuclis i usant el nucli $G^{6\times3\times1}$ a la primera (l'òptim en un tensor $\chi^{288\times7\times3}$ i 100 mostres de ràfega).

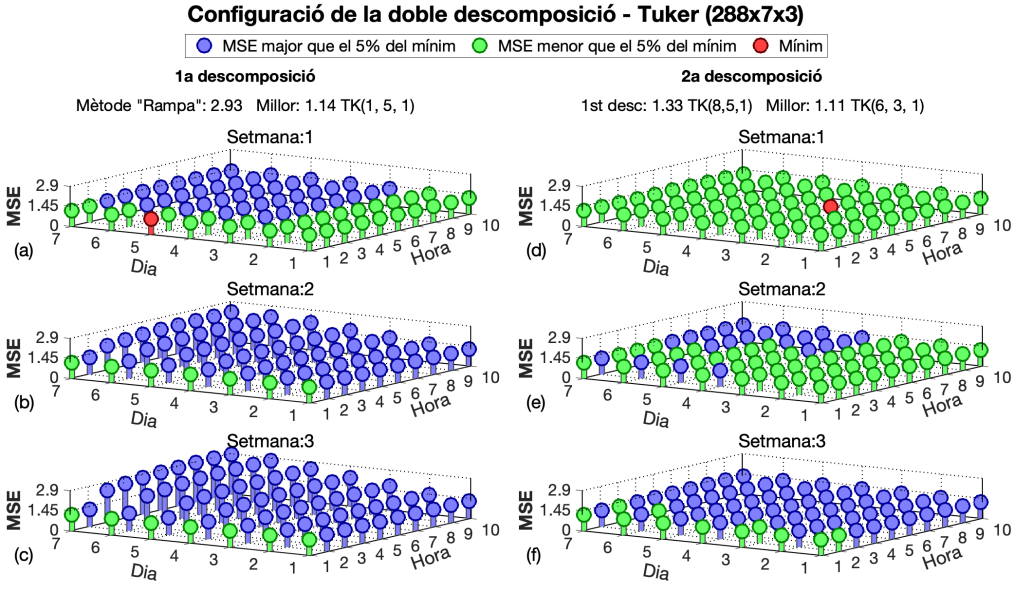

Figura 6.26: Configuració de la doble descomposició amb model Tucker, tensor $\chi^{288\times7\times3}$ i ràfega de 200. (a) MSE de la primera descomposició per diferents nuclis. (b) MSE de la segona descomposició per varis nuclis i usant el nucli $G^{8\times5\times1}$ a la primera (l'òptim en un tensor $\chi^{288\times7\times7}$ i 100 mostres de ràfega).

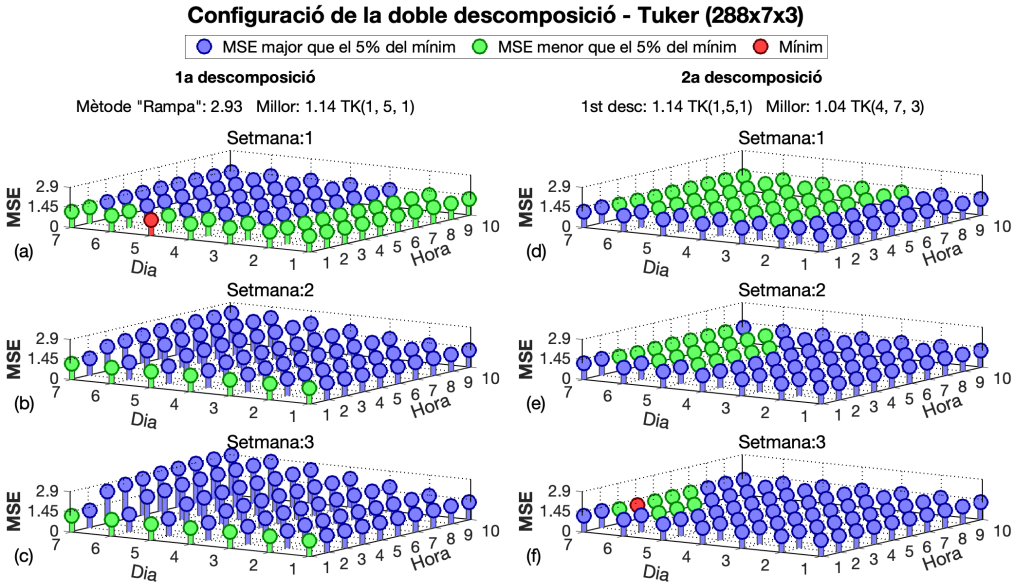

Figura 6.27: Configuració de la doble descomposició amb model Tucker, tensor $\chi^{288\times7\times3}$ i ràfega de 200. (a) MSE de la primera descomposició per diferents nuclis. (b) MSE de la segona descomposició per varis nuclis i usant el nucli $G^{1\times5\times1}$ a la primera (l'òptim en aquestes condicions de tensor i mida de ràfega).

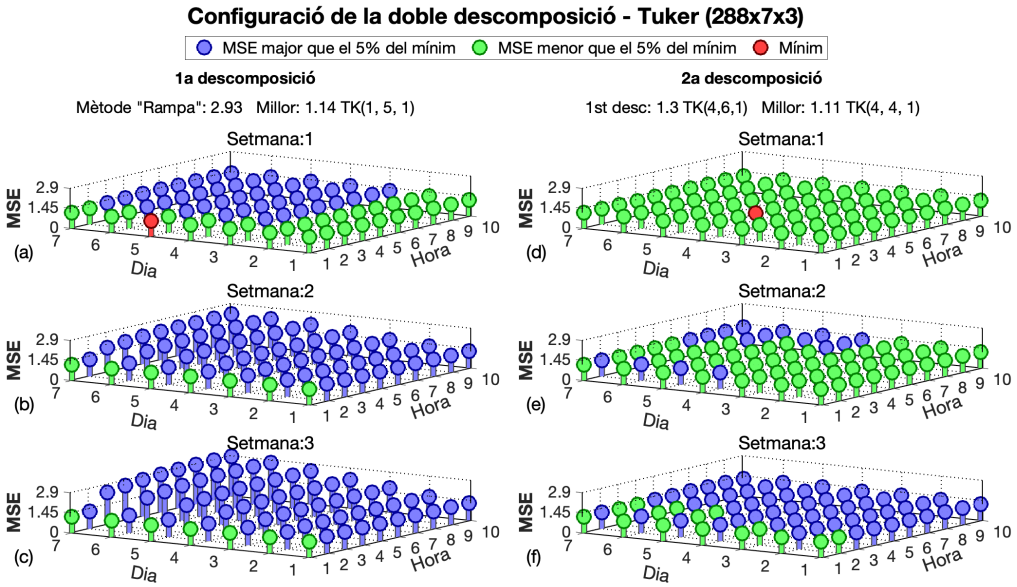

Figura 6.28: Configuració de la doble descomposició amb model Tucker, tensor $\chi^{288\times7\times3}$ i ràfega de 200. (a) MSE de la primera descomposició per diferents nuclis. (b) MSE de la segona descomposició per varis nuclis i usant el nucli $G^{4\times6\times1}$ a la primera (l'òptim en un tensor $\chi^{288\times7\times7}$ i 200 mostres de ràfega).

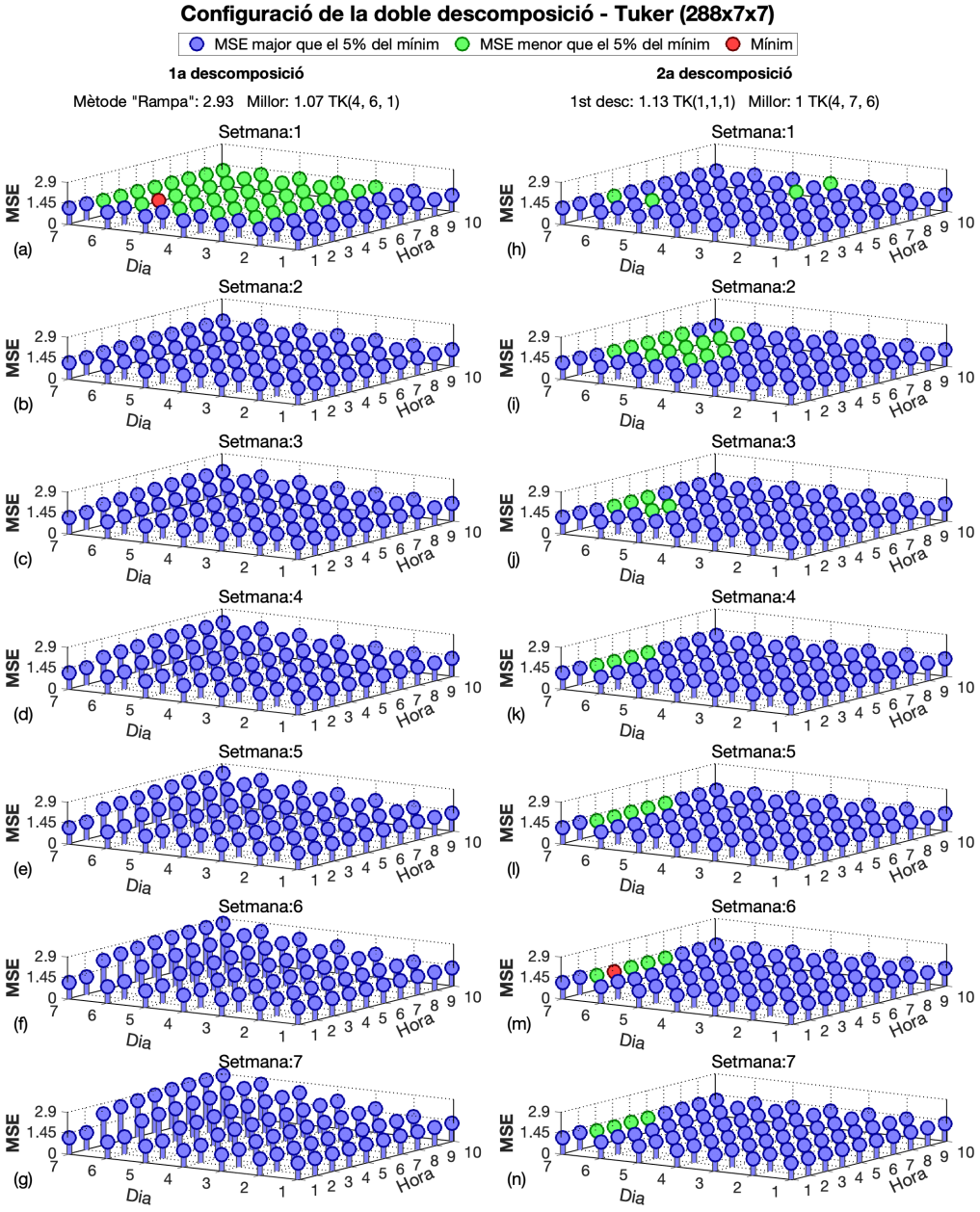

Figura 6.29: Configuració de la doble descomposició amb model Tucker, tensor $\chi^{288\times7\times7}$ i ràfega de 200. (a) MSE de la primera descomposició per diferents nuclis. (b) MSE de la segona descomposició per varis nuclis i usant el nucli $G^{1\times1\times1}$ a la primera (el nucli més simple possible).

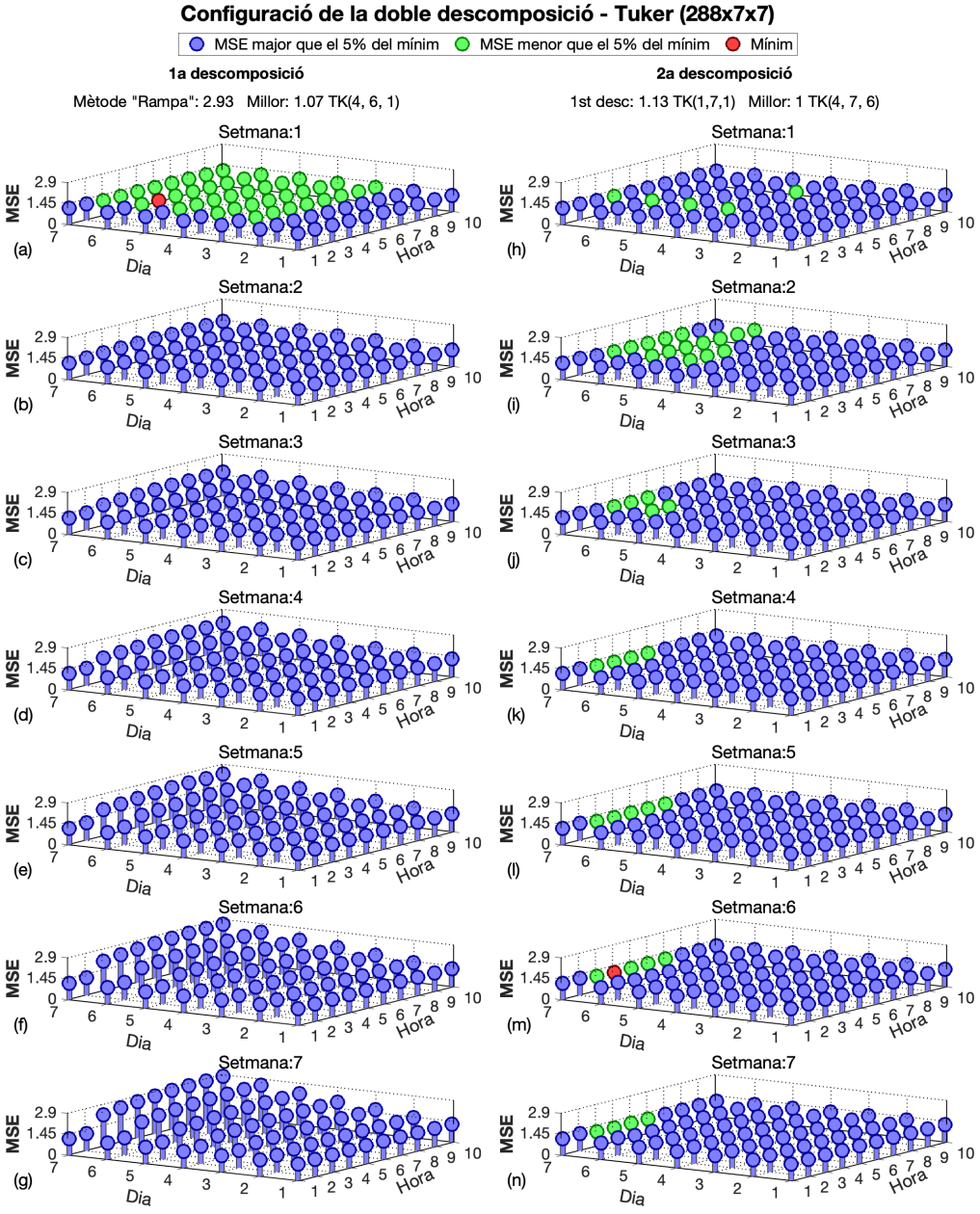

Figura 6.30: Configuració de la doble descomposició amb model Tucker, tensor $\chi^{288\times7\times7}$ i ràfega de 200. (a) MSE de la primera descomposició per diferents nuclis. (b) MSE de la segona descomposició per varis nuclis i usant el nucli $G^{1\times7\times1}$ a la primera (un nucli que simplifica hores i setmanes però manté la mida de la dimensió setmanal).

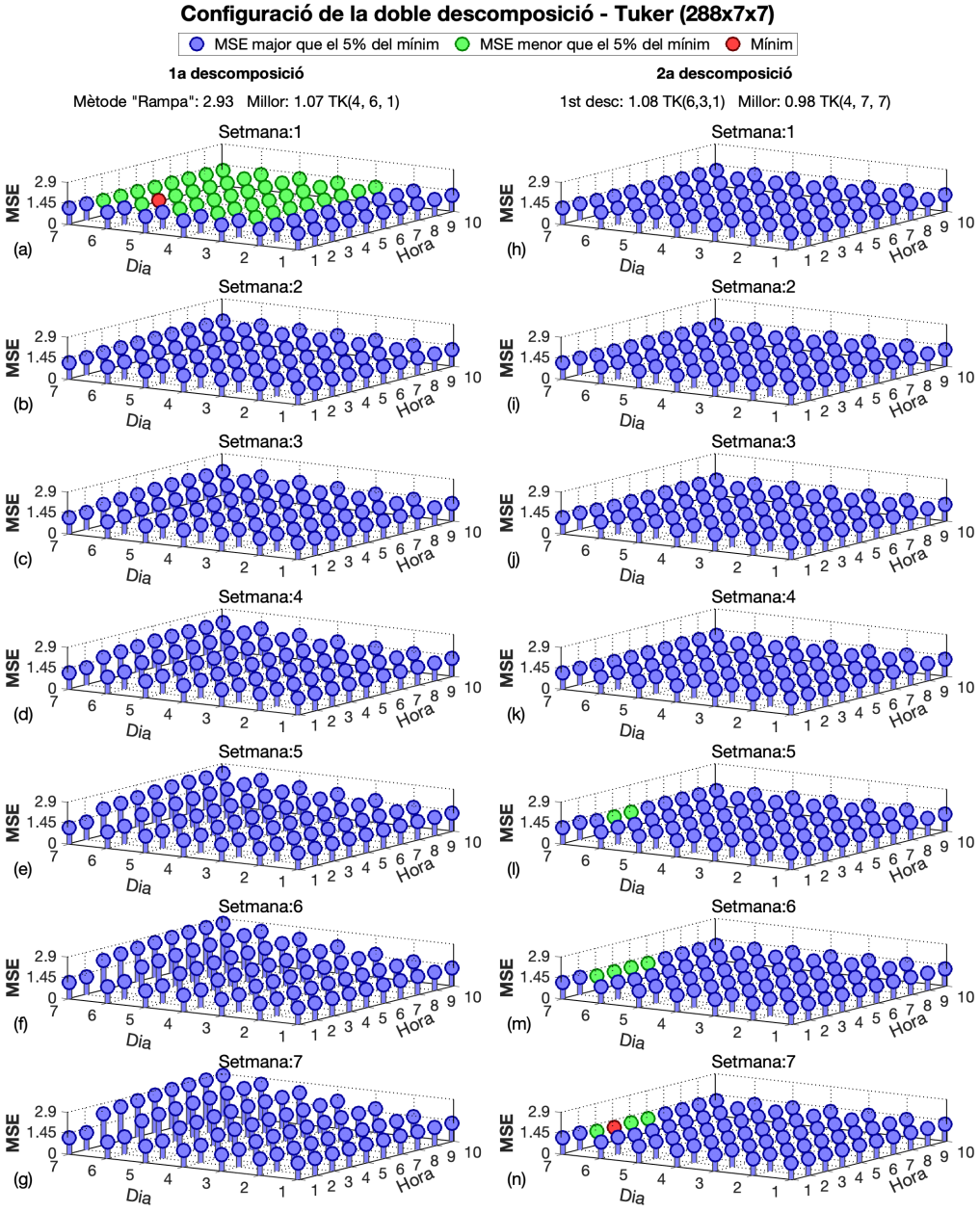

Figura 6.31: Configuració de la doble descomposició amb model Tucker, tensor $\chi^{288\times7\times7}$ i ràfega de 200. (a) MSE de la primera descomposició per diferents nuclis. (b) MSE de la segona descomposició per varis nuclis i usant el nucli $G^{6\times3\times1}$ a la primera (l'òptim en un tensor $\chi^{288\times7\times3}$ i 100 mostres de ràfega).

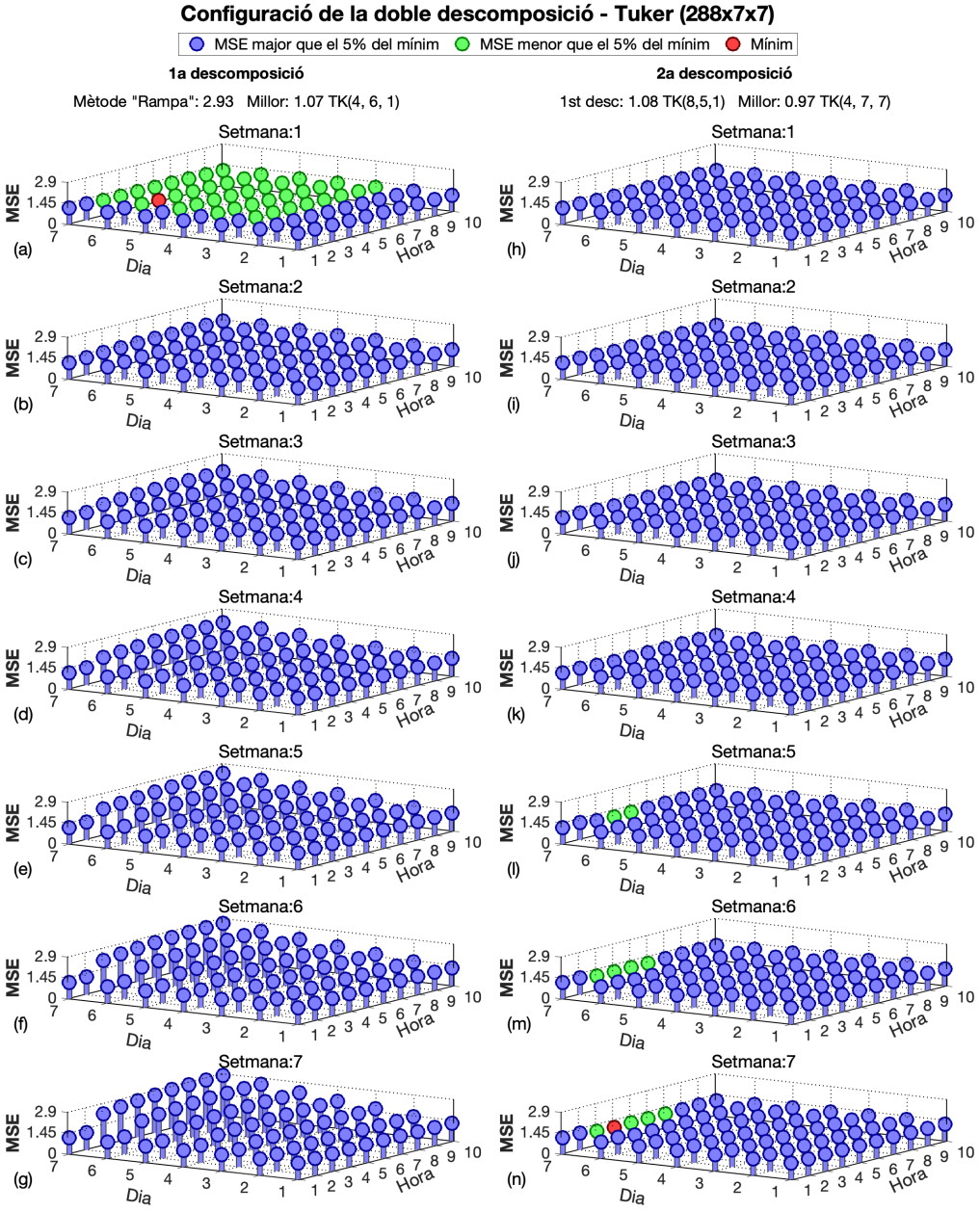

Figura 6.32: Configuració de la doble descomposició amb model Tucker, tensor $\chi^{288\times7\times7}$ i ràfega de 200. (a) MSE de la primera descomposició per diferents nuclis. (b) MSE de la segona descomposició per varis nuclis i usant el nucli $G^{8\times5\times1}$ a la primera (l'òptim en un tensor $\chi^{288\times7\times7}$ i 100 mostres de ràfega).

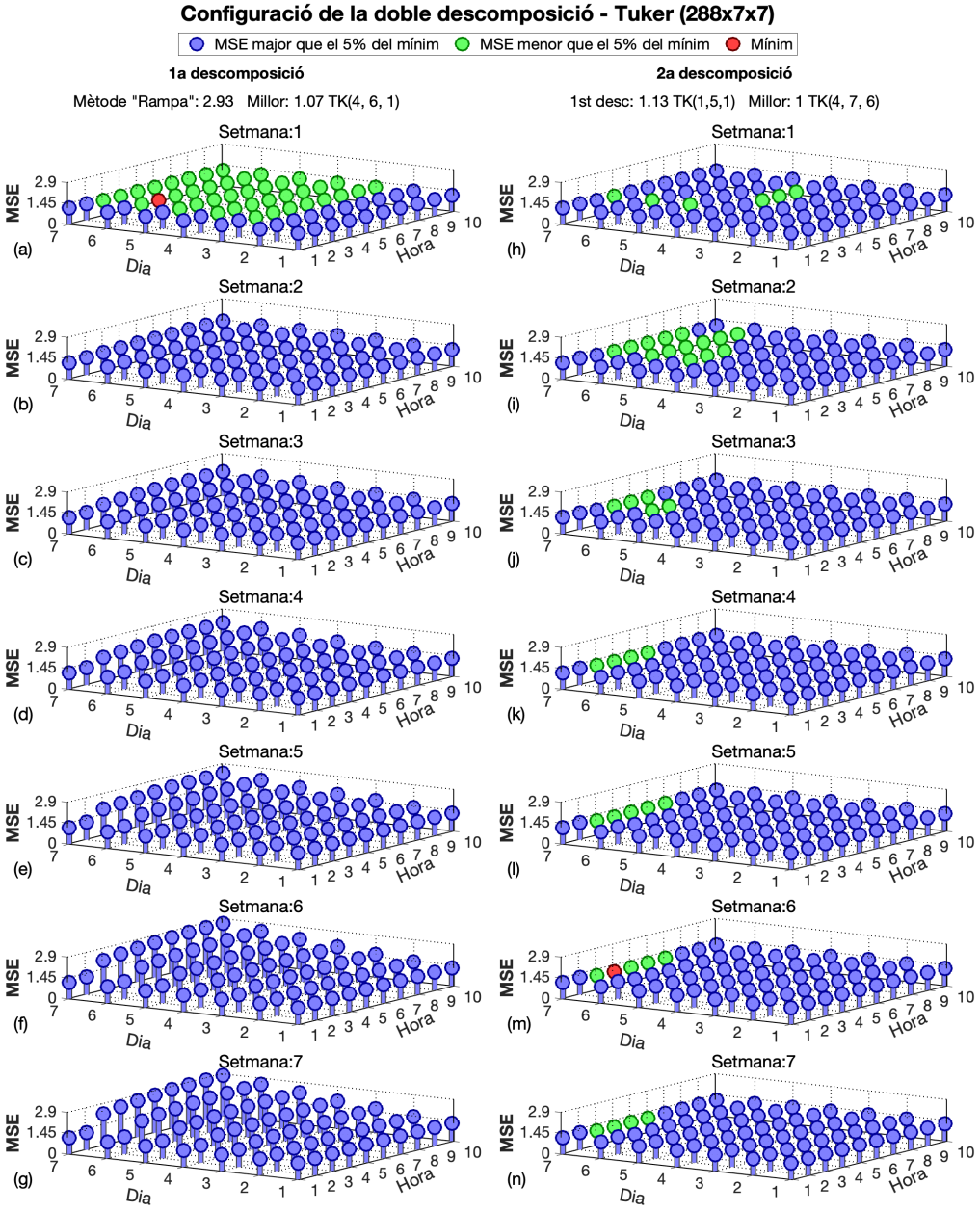

Figura 6.33: Configuració de la doble descomposició amb model Tucker, tensor $\chi^{288\times7\times7}$ i ràfega de 200. (a) MSE de la primera descomposició per diferents nuclis. (b) MSE de la segona descomposició per varis nuclis i usant el nucli $G^{1\times5\times1}$ a la primera (l'òptim en un tensor $\chi^{288\times7\times3}$ i 200 mostres de ràfega).

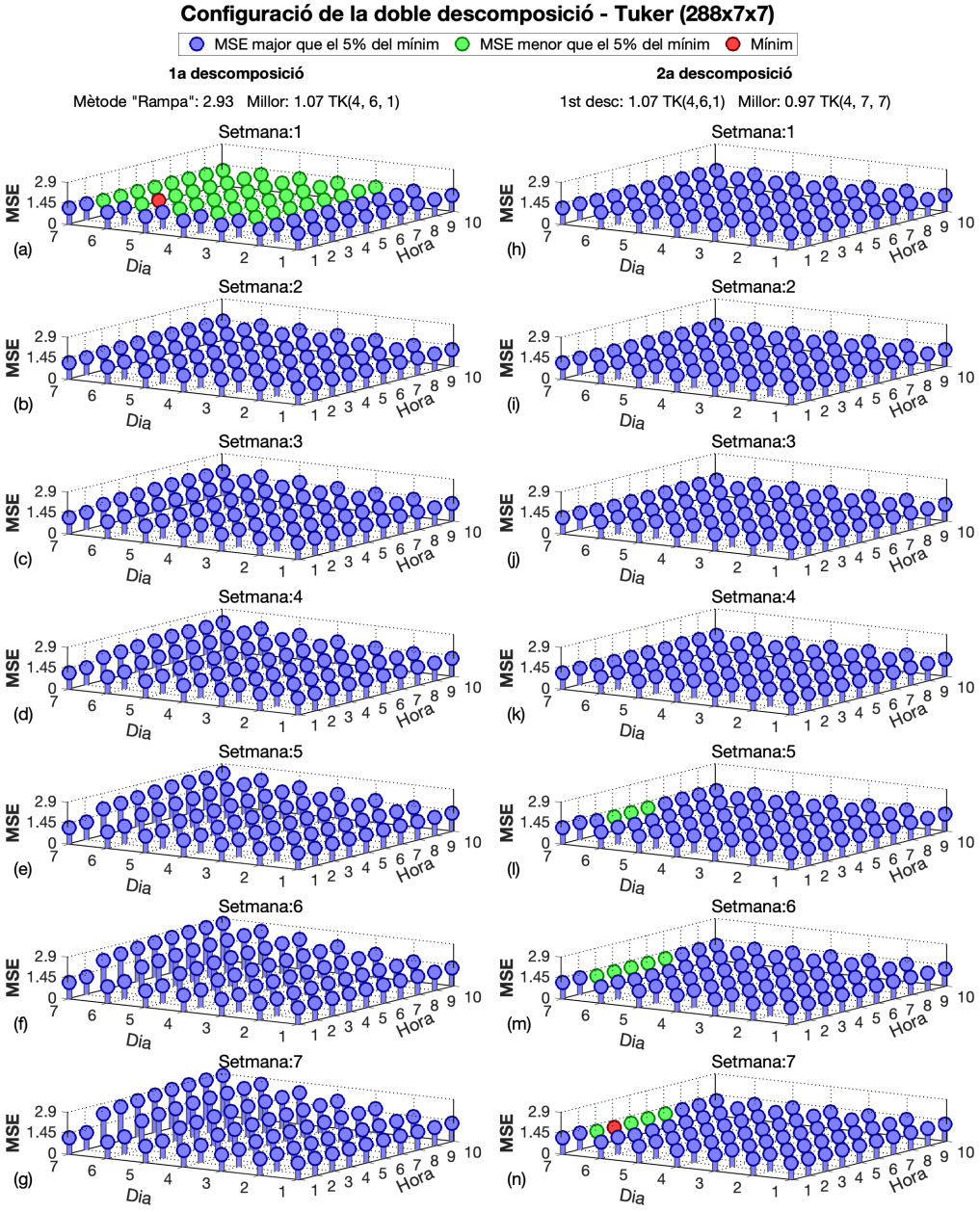

Figura 6.34: Configuració de la doble descomposició amb model Tucker, tensor $\chi^{288\times7\times7}$ i ràfega de 200. (a) MSE de la primera descomposició per diferents nuclis. (b) MSE de la segona descomposició per varis nuclis i usant el nucli $G^{4\times6\times1}$ a la primera (l'òptim en aquestes condicions de tensor i mida de ràfega).

A la figures 6.35-6.44 es mostren uns exemples del senyal reconstruït amb el procés de la descomposició doble. S'utilitza la millor configuració de nuclis pel cas del model Tucker i una ràfega de 200 mostres, $G^{4\times6\times1}$ per la primera descomposició i $G^{4\times7\times7}$ per la segona.

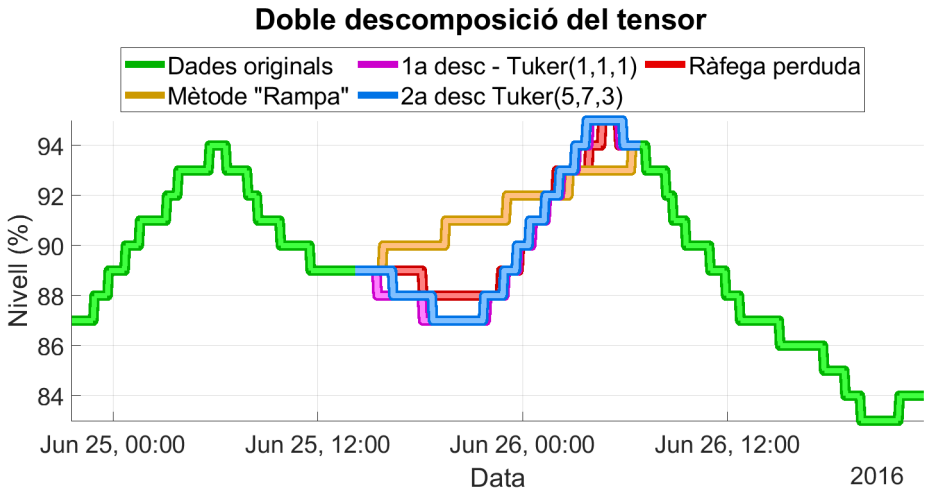

Figura 6.35: Exemple del mètode de reconstrucció de dades amb doble descomposició del model Tucker. La línia verda mostra el senyal original del sensor de nivell. La taronja mostra l'estimació amb el mètode lineal "Rampa". La lila mostra el resultat de la primera descomposició amb nucli $G^{4\times6\times1}$. La blava mostra el resultat de la segona descomposició amb $G^{4\times7\times7}$.

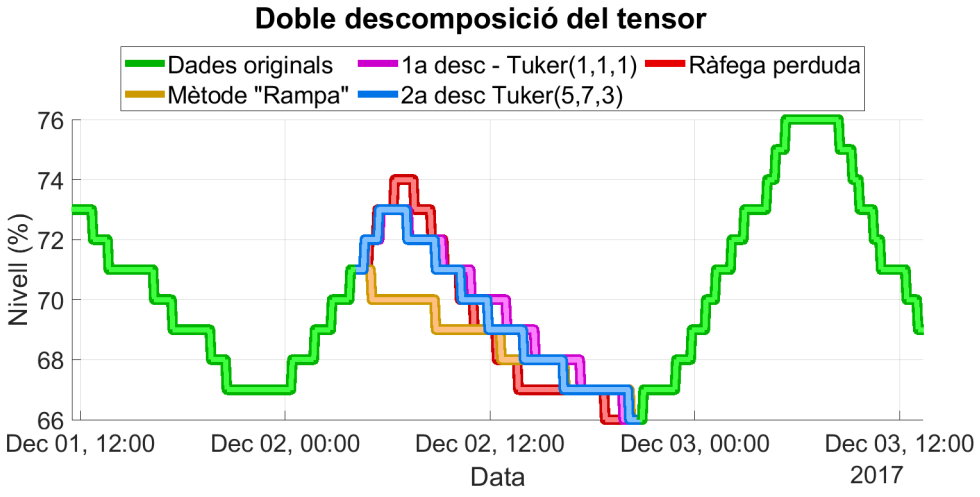

Figura 6.36: Exemple del mètode de reconstrucció de dades amb doble descomposició del model Tucker. La línia verda mostra el senyal original del sensor de nivell. La taronja mostra l'estimació amb el mètode lineal "Rampa". La lila mostra el resultat de la primera descomposició amb nucli $G^{4\times6\times1}$. La blava mostra el resultat de la segona descomposició amb $G^{4\times7\times7}$.

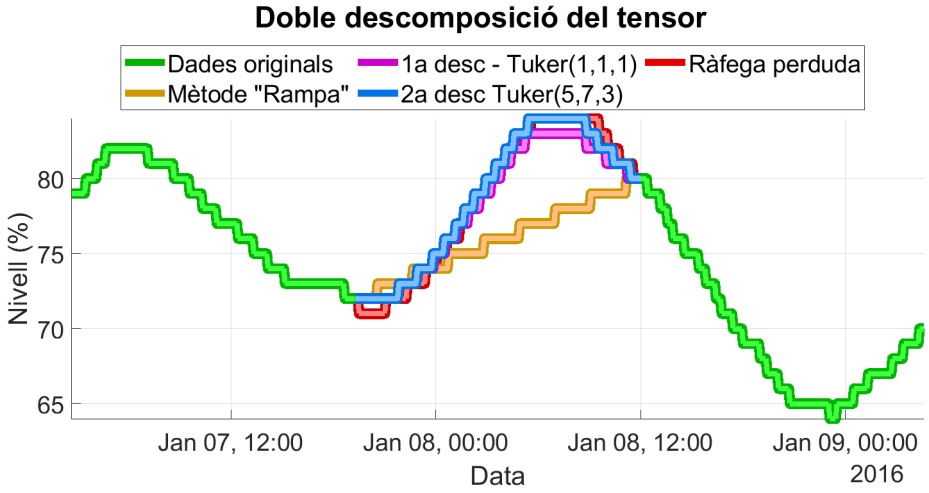

Figura 6.37: Exemple del mètode de reconstrucció de dades amb doble descomposició del model Tucker. La línia verda mostra el senyal original del sensor de nivell. La taronja mostra l'estimació amb el mètode lineal "Rampa". La lila mostra el resultat de la primera descomposició amb nucli $G^{4\times6\times1}$. La blava mostra el resultat de la segona descomposició amb $G^{4\times7\times7}$.

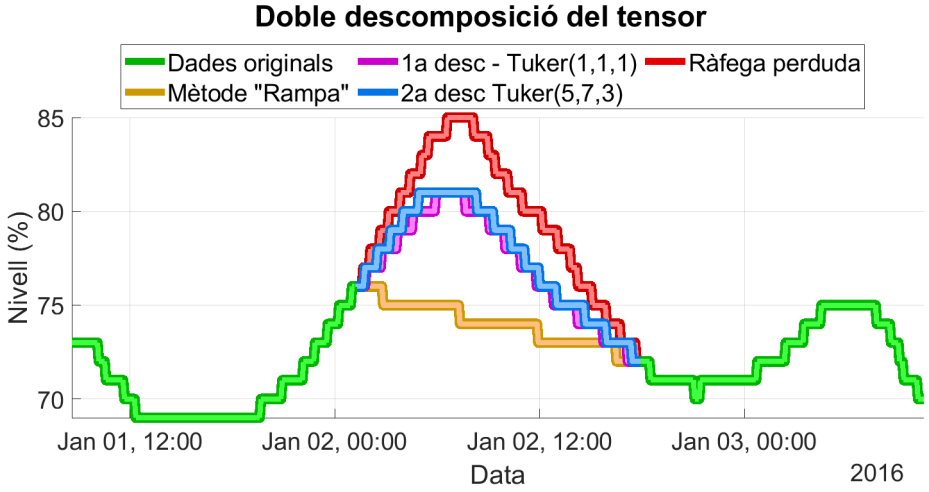

Figura 6.38: Exemple del mètode de reconstrucció de dades amb doble descomposició del model Tucker. La línia verda mostra el senyal original del sensor de nivell. La taronja mostra l'estimació amb el mètode lineal "Rampa". La lila mostra el resultat de la primera descomposició amb nucli $G^{4\times6\times1}$. La blava mostra el resultat de la segona descomposició amb $G^{4\times7\times7}$.

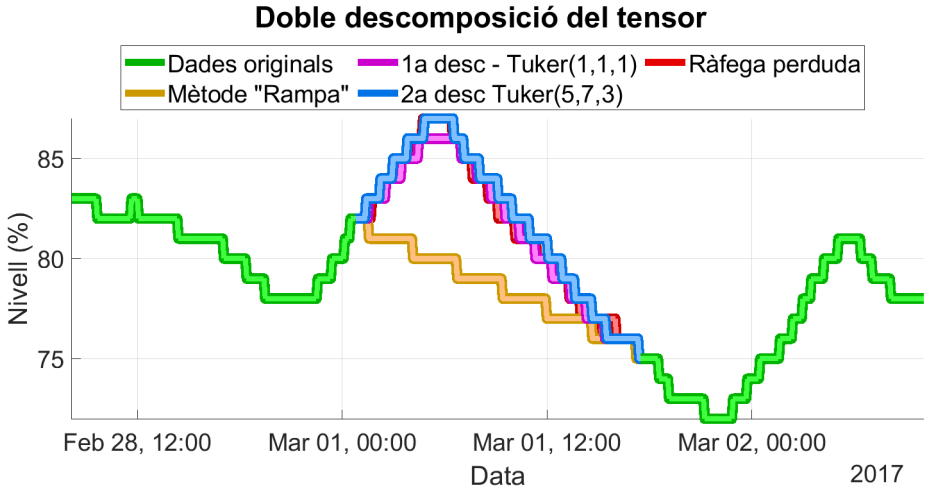

Figura 6.39: Exemple del mètode de reconstrucció de dades amb doble descomposició del model Tucker. La línia verda mostra el senyal original del sensor de nivell. La taronja mostra l'estimació amb el mètode lineal "Rampa". La lila mostra el resultat de la primera descomposició amb nucli $G^{4\times6\times1}$. La blava mostra el resultat de la segona descomposició amb $G^{4\times7\times7}$.

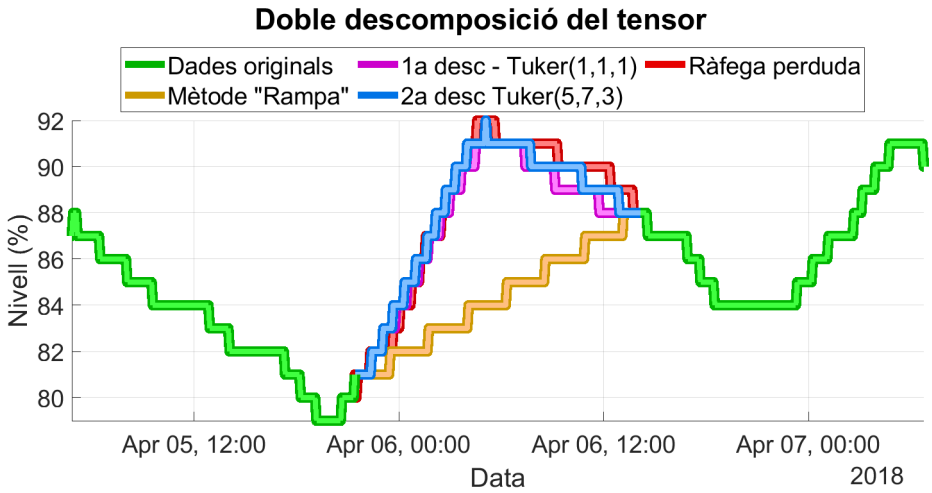

Figura 6.40: Exemple del mètode de reconstrucció de dades amb doble descomposició del model Tucker. La línia verda mostra el senyal original del sensor de nivell. La taronja mostra l'estimació amb el mètode lineal "Rampa". La lila mostra el resultat de la primera descomposició amb nucli $G^{4\times6\times1}$. La blava mostra el resultat de la segona descomposició amb $G^{4\times7\times7}$.

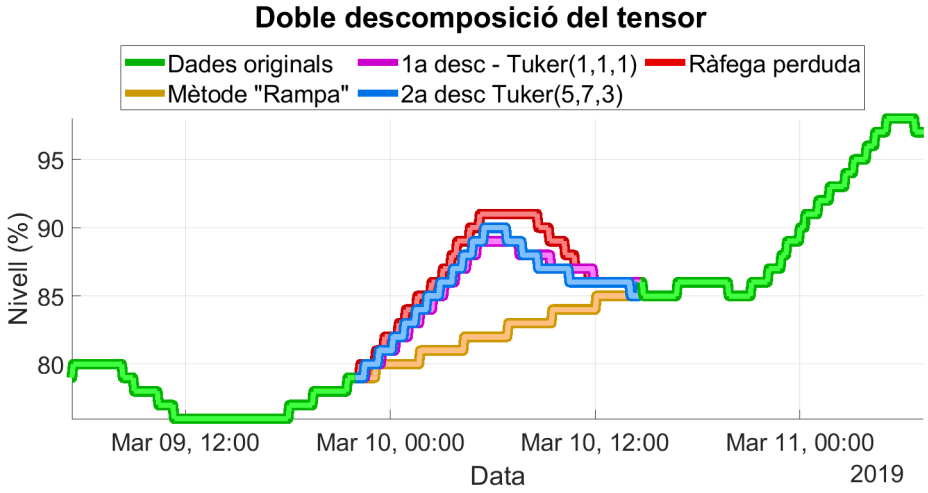

Figura 6.41: Exemple del mètode de reconstrucció de dades amb doble descomposició del model Tucker. La línia verda mostra el senyal original del sensor de nivell. La taronja mostra l'estimació amb el mètode lineal "Rampa". La lila mostra el resultat de la primera descomposició amb nucli $G^{4\times6\times1}$. La blava mostra el resultat de la segona descomposició amb $G^{4\times7\times7}$.

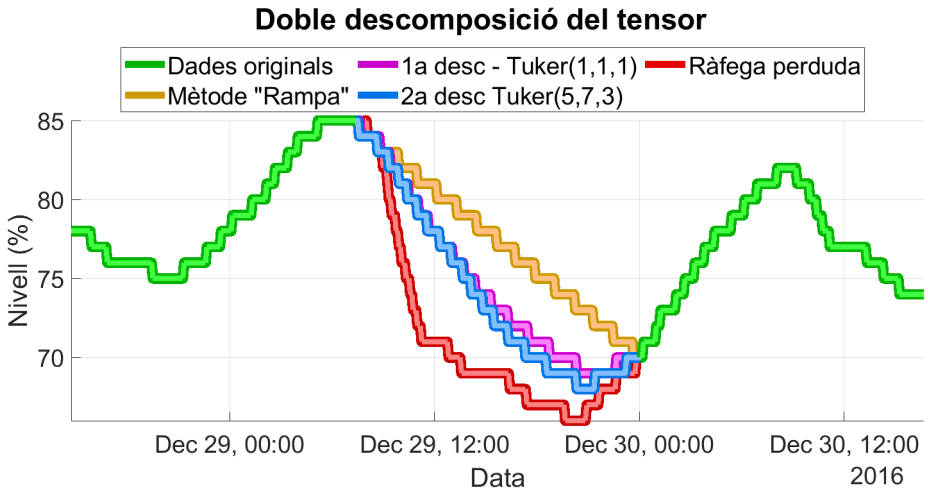

Figura 6.42: Exemple del mètode de reconstrucció de dades amb doble descomposició del model Tucker. La línia verda mostra el senyal original del sensor de nivell. La taronja mostra l'estimació amb el mètode lineal "Rampa". La lila mostra el resultat de la primera descomposició amb nucli $G^{4\times6\times1}$. La blava mostra el resultat de la segona descomposició amb $G^{4\times7\times7}$.

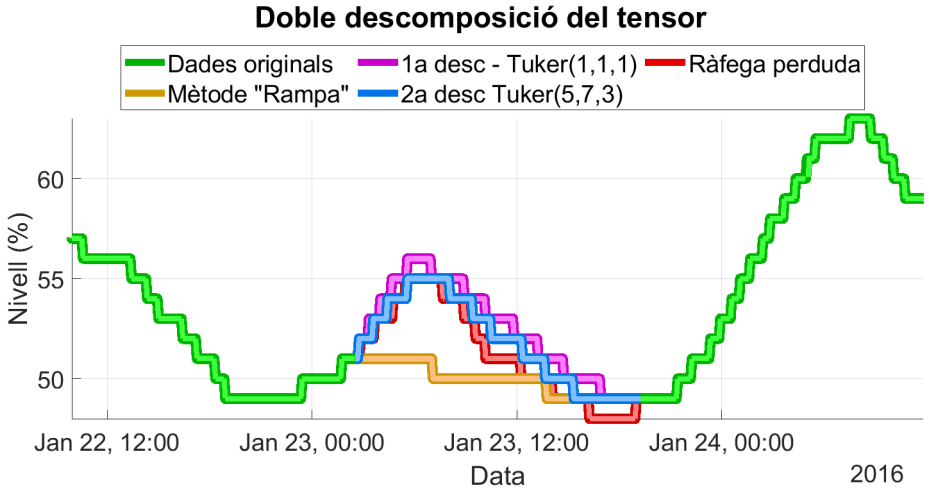

Figura 6.43: Exemple del mètode de reconstrucció de dades amb doble descomposició del model Tucker. La línia verda mostra el senyal original del sensor de nivell. La taronja mostra l'estimació amb el mètode lineal "Rampa". La lila mostra el resultat de la primera descomposició amb nucli $G^{4\times6\times1}$. La blava mostra el resultat de la segona descomposició amb $G^{4\times7\times7}$.

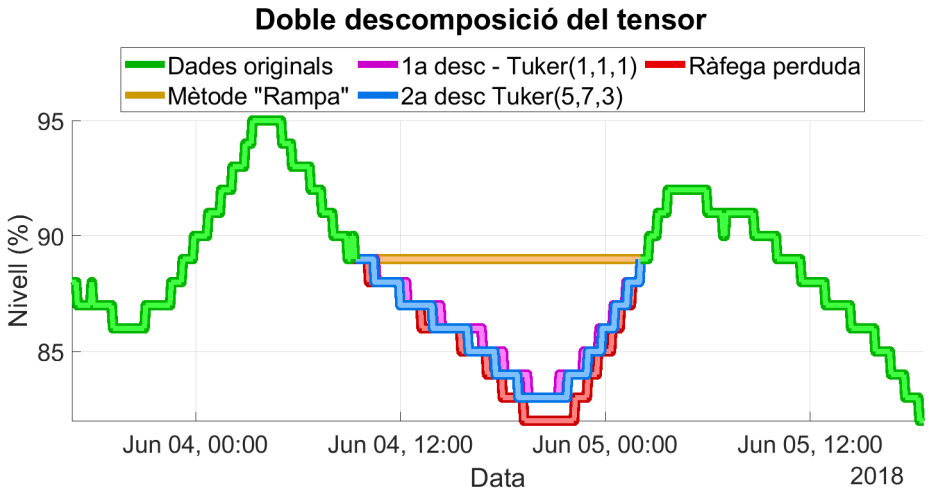

Figura 6.44: Exemple del mètode de reconstrucció de dades amb doble descomposició del model Tucker. La línia verda mostra el senyal original del sensor de nivell. La taronja mostra l'estimació amb el mètode lineal "Rampa". La lila mostra el resultat de la primera descomposició amb nucli $G^{4\times6\times1}$. La blava mostra el resultat de la segona descomposició amb $G^{4\times7\times7}$.

6.4. Resultats

Per tal d'avaluar les diferents millores proposades per refinar la metodologia tensorial es realitzen un seguit de simulacions, aplicant cada una de les millores per separat i aplicant-les de forma combinada, figura 6.45.

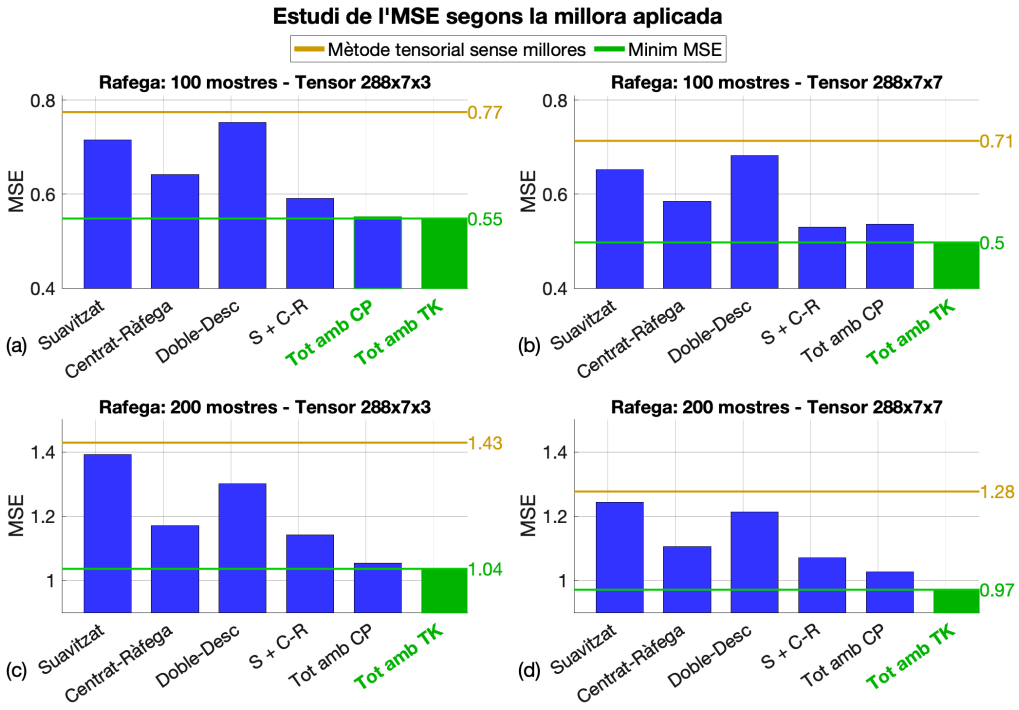

Figura 6.45: MSE segons les diferents millores aplicades. Les simulacions es fan pels tensors $\chi^{288\times7\times3}$ i $\chi^{288\times7\times7}$ i amb 100 i 200 mostres de ràfega. La línia taronja indica l'MSE en cas de no aplicar cap de les millores. "Data smoothing" és el cas d'aplicar el suavitzat del senyal. "Burst centered tensorization" el cas del re-ordenament del tensor segons la posició de la ràfega. "Double decom" el cas de la doble descomposició amb $G^{4\times6\times1}$ i $G^{4\times7\times7}$ pel model Tuker i $G^{1\times1\times1}$ i $G^{15\times15\times15}$ pel CP. "Ds-Bc" és la combinació de suavitzat i re-ordenament sense aplicar la descomposició doble. "All with CP" i "All with TK"són els casos d'aplicar totes les millores pels models CP i Tucker respectivament.

Realitzant la modificació en la manera de col·locar les dades dins del tensor de la secció 6.1 s'obté una millora substancial respecte l'anterior manera. A més aquesta millora es manté proporcionalment, respecte a la utilització de diferents tensors i la restauració de diferents mides de ràfega.

La tècnica de suavitzar el senyal de la secció 6.2 millora lleugerament els resultats en tots els casos de mida de tensor i ràfega provats.

Per comprovar que la millora proposada a la secció 6.3 és fiable i robusta es calcula l'MSE amb diferents configuracions per la primera descomposició com es mostra a la taula 6.1 i a les figures 6.7-6.34. A la taula en concret, es seleccionen les configuracions d'error mínim de les figures anteriors (primera columna de les gràfiques de les figures). O sigui els òptims segons les proves realitzades per cada mida de tensor i ràfega, CP(1), TK(6,3,1), TK(8,5,1), TK(1,5,1) i TK(4,6,1). A la taula 6.1 i a les figures 6.7-6.34 es fa palesa la consistència de l'algoritme, ja que amb les proves realitzades amb diferents nuclis per la primera descomposició, els resultats sempre milloren, i en més o menys mesura s'acosten al mínim possible.

L'MSE corresponent a 100 mostres perdudes baixa de 0,71 (el millor resultat obtingut al no aplicar cap de les millores) fins a 0,50 (el millor amb les millores incorporades). Això significa un 39,5 % de reducció de l'MSE. En el cas de 200 mostres de ràfega baixa de 1,28 a 0,97, aproximadament un 24,2 %.

Tabla 6.1: MSE de les simulacions amb 100 i 200 mostres de ràfega, B, i utilitzant els tensors de 3 i 7 setmanes, n_w. En els casos presentats de descomposició doble es proven uns nuclis concrets per la primera descomposició (els que han obtingut els millors resultats en els experiments,) i es mostra per cada un d'ells, el millor resultat de tots els nuclis provats per la segona descomposició.

MSE	$B = 100$ $n_w = 3$	$B = 100$ $n_w = 7$	$B = 200$ $n_w = 3$	$B = 200$ $n_w = 7$
Mètode sense millores (descomposició)				
CP òptima	0.87	0.80	1.70	1.58
TK òptima	0.77	0.71	1.43	1.28
Mètode amb millores (1a i 2a descomposicions)				
1a: CP(1), 2a: CP òptima	0.55	0.54	1.05	1.03
1a: TK(6,3,1), 2a: CP òptima	0.57	0.52	1.14	1.02
1a: TK(8,5,1), 2a: CP òptima	0.57	0.53	1.14	1.03
1a: TK(1,5,1), 2a: CP òptima	0.55	0.53	1.06	1.02
1a: TK(4,6,1), 2a: CP òptima	0.56	0.53	1.06	1.02
1a: TK(1,1,1), 2a: TK òptima	0.54	0.52	1.04	1.00
1a: TK(6,3,1), 2a: TK òptima	0.55	0.51	1.11	0.98
1a: TK(8,5,1), 2a: TK òptima	0.54	0.50	1.11	0.97
1a: TK(1,5,1), 2a: TK òptima	0.53	0.52	1.04	1.00
1a: TK(4,6,1), 2a: TK òptima	0.55	0.50	1.11	0.97

6.5. Conclusions

Analitzant els resultats obtinguts respecte a les millores proposades es determina que el procediment que influeix més positivament és el re-ordenament del tensor segons la posició de la ràfega descrit a la secció 6.1.

Un resultat curiós és el de la doble descomposició de la secció 6.3, ja que sembla que en proporció és més efectiu quan s'utilitza en combinació amb les altres millores. Es pot observar comparant la reducció de l'MSE que suposa aplicar només la descomposició doble amb la que s'obté aplicant només la combinació de re-ordenament i suavitzat, i amb la que s'obté utilitzant totes les millores proposades, especialment en el cas del model Tucker.

Amb qualsevol de les condicions de mides de ràfega i tensor l'efecte de cada una de les millores proposades és complementari, és a dir, que cada una de les millores contribueix en reduir l'MSE. De fet, aplicar totes les millores suposa una reducció important de l'MSE en comparació en no fer-ne servir cap, en totes les condicions provades. El nucli òptim de les descomposicions tensorials és diferent segons les condicions de mida de tensor i de ràfega utilitzats, però amb característiques similars com es pot apreciar a les figures 6.7-6.34 amb zones verdes semblants.

La millora aconseguida gràcies a les modificacions proposades en aquest capítol, respecte a la metodologia inicial de la secció 5.2 de restauració de dades amb mètodes tensorials, està aproximadament entre el 40 i el 25 %, segons la mida de la ràfega de dades, 100 i 200 mostres respectivament.

Capítol 7

CONCLUSIONS

Actualment la recol·lecció i l'acumulació de dades ha fet un salt endavant, gràcies a la gran varietat de dispositius i sensors capaços de transmetre dades pràcticament des de qualsevol lloc i en qualsevol moment, i a l'augment de la capacitat de guardar grans quantitats d'informació. De fet, s'ha arribat a un punt en què es guarden moltes més dades de les que realment es fan servir. En el moment de processar tota aquesta informació surt a la llum el problema de les dades perdudes que cal tractar. L'àmbit de les xarxes de distribució d'aigua i en general dels recursos hidrològics no en són una excepció, [51–54]. El problema de la pèrdua de dades es complica quan es produeix en forma de ràfegues llargues, és a dir, quan es perden conjunts grans de mostres consecutives.

Amb les dades proporcionades per Aigües de Vic del sensor de nivell del Castell d'en Plantes, el dipòsit principal de la ciutat de Vic, s'han realitzat un seguit de simulacions consistents en generar ràfegues de mostres perdudes de forma aleatòria a un historial fiable, és a dir, del que se n'ha verificat la integritat de les dades. Aquestes ràfegues perdudes simuladament es restauren mitjançant diferents mètodes de reconstrucció de dades i els resultats s'avaluen comparant les estimacions obtingudes amb els valors reals de l'historial verificat. D'aquesta forma es pot establir una valoració quantitativa de tots els mètodes que s'han provat, tant de mètodes existents com de mètodes proposats.

En el context del problema tractat, pel que fa a la mida de les ràfegues perdudes, s'ha observat que, per a ràfegues més petites de 25 mostres tots els mètodes provats generen errors similars i que es poden considerar petits. És a dir, que s'aconsegueix estimar correctament el senyal original. A mesura que augmenta la longitud de la ràfega, els mètodes més complexes obtenen millors resultats, figura 5.23. El mètode més simple, el "Dada anterior" que fan servir els operaris (secció 4.1), empitjora dràsticament els resultats de seguida que la ràfega augmenta de 25 mostres. El mètode "FIR" de la secció 4.3, proposat com a mètode lineal mitjançant dos versions del predictor de Wiener mostra una lleugera millora respecte del mètode "Rampa" (el millor dels mètodes utilitzats pels operaris, secció 4.2). No obstant, a mesura que les ràfegues són de més

de 100 mostres, pot proporcionar reconstruccions del senyal força diferents a la realitat. D'acord als resultats obtinguts els mètodes basats en tensors de la secció 5.2 comencen a superar de forma clara als millors mètodes lineals a partir de longituds de ràfegues de 50 mostres.

En aquesta tesi, inicialment, es restauren les dades mitjançant mètodes lineals. La proposta en aquest sentit és utilitzar dos filtres predictors que es combinen per reconstruir el senyal. Un dels filtres actuant en la direcció temporal (avançant en el temps per omplir la ràfega) i l'altre en la direcció inversa (retrocedint en el temps per omplir la ràfega). Els coeficients d'aquests filtres es calculen d'acord al mètode de Wiener, cosa que implica estimar les autocorrelacions del senyal. Aquest és un punt crític d'aquest mètode, ja que s'ha observat que la finestra utilitzada (és a dir, la quantitat de mostres de l'historial de la senyal que es fan servir per fer l'estimació dels coeficients d'autocorrelació) i l'ordre del filtre, tenen un impacte important en el funcionament del mètode. Es fa difícil determinar quins són els valors òptims de la finestra i de l'ordre del filtre, ja que poden variar segons el tram del senyal que cal reconstruir. A més l'estimació de la la matriu d'autocorrelació requereix força temps de càlcul, que cal repetir cada cop que es reconstrueix una ràfega. Al final, el mètode proposat funciona de forma similar als mètodes sofisticats que utilitzen interpolació en alguna de les seves formes. La principal limitació que troba aquest mètode, comú a tots els mètodes lineals explorats, és que a mesura que les ràfegues perdudes s'allarguen, superant les 100 mostres, les reconstruccions de senyal obtingudes empitjoren i difereixen dels valors reals de la senyal. Tot i això, en cas de ràfegues llargues, el mètode proposat conserva molt millor la forma del senyal que els mètodes que fan servir els operaris, que de seguida la deformen com s'aprecia a la figura 4.23.

L'observació del senyal de nivell estudiat mostra que hi ha certa regularitat, certs patrons, en escales diàries i també setmanals. Es coneix que hi ha regularitats a l'hora d'omplir i buidar els dipòsits durant el dia ja que els operaris segueixen certes rutines i els consumidors (tant particulars com empreses) també. Aquestes regularitats també tenen un efecte a nivell setmanal, ja que tant particulars com empreses tenen un comportament diferent els caps de setmana, sobretot pel que fa als particulars. S'esperava que el filtre de Wiener seria capaç d'aprofitar aquestes periodicitats amagades, però no ho aconseguix amb prou eficàcia.

Per tal de treure més profit de les regularitats, difícils de veure a simple vista,

es proposa la utilització dels tensors. Per fer-ho, el primer pas va ser introduir l'historial de dades del sensor de nivell en un tensor. Això significa organitzar un senyal unidimensional en un tensor de vàries dimensions, de forma que les dimensions extra generades ajudin a aprofitar les periodicitats del senyal. Al final s'ha observat com l'àlgebra tensorial permet superar algunes limitacions de l'àlgebra vectorial i matricial ja que el mètode basat en tensors ha resultat molt més eficaç que els mètodes lineals.

La progressió en el desenvolupament dels algorismes ha anat paral·lela a la progressió de la tesi. El primer mètode proposat, basat en tensors, refina les estimacions fetes amb mètodes lineals senzills. De l'observació acurada de cada un dels passos que s'hi realitzen, s'han pogut identificar problemes tals com el de la pèrdua de continuïtat o l'esglaonat del senyal, per posar-hi solució. El resultat és un primer mètode ad hoc que supera les prestacions dels mètodes basats en tensors que s'han provat. En el primer article publicat en una revista indexada, apèndix B, es presenta aquest mètode i es mostren els seus resultats.

Després d'aquesta publicació, la primera millora important que es va incorporar al mètode inicial es va publicar al congrés CCIA 2019, àpendixC. En aquest article es mostra una nova forma d'organitzar les dades en el tensor, mantenint la mida i la filosofia del tensor original, però modificant la manera com es col·loca la ràfega de dades perdudes al tensor, en concret forçant que es situï al seu cor, el que equival a la posició central en cada una de les seves dimensions.

A més es van identificar altres millores, de manera que al final de la tesi s'ha pogut fer evolucionar el mètode i obtenir un nou algorisme, actualment en fase de publicació. Aquest nou algorisme, per una banda millora els resultats del publicat anteriorment, i per l'altra proporciona major estabilitat, ja que depèn de pocs paràmetres que es poden generalitzar obtenint resultats similars i satisfactoris. Aquest segon algorisme fa servir una doble descomposició tensorial, tal com s'explica a la secció 6.3 del capítol 6. Per a ràfegues de 200 mostres la millora que obté sobre el primer algorisme proposat, en termes de l'MSE és de quasi un 40 %.

Per tant, el procediment presentat permet millorar l'estimació d'una ràfega de dades perduda respecte als mètodes previs disponibles, fent servir un conjunt de dades anteriors i posteriors a la ràfega, un tensor de tres dimensions tem-

porals, dos descomposicions tensorials i un procés per mantenir la continuïtat del senyal. En concret, fent servir aquest mètode, l'MSE (per mostra) obtingut amb el mètode lineal que fan servir els operaris d'Aigües de Vic, pot arribar a baixar de 1,2 a 0,5 en el cas de ràfegues de 100 mostres (un 58 %) i de 2,93 a 0,97 en les ràfegues de 200 mostres (un 66 %). Al observar les restauracions del senyal d'aquest algoritme, representant-les conjuntament amb els valors del senyal original, reconforta veure com s'assemblen, així com, que la tendència del senyal original es reflecteixi amb eficàcia en les reconstruccions realitzades.

Per tal d'investigar el comportament del mètode proposat es realitzen proves amb dos dels models de descomposició tensorials més usats, el CP i el Tucker. Les dues descomposicions actuen de forma força similar, però sembla que la Tuker, al poder configurar més paràmetres, permet obtenir resultats lleugerament millors. Tot i això la millora és considerable amb els dos models i en el cas del model CP, el fet que la configuració sigui més simple, també la fa resultar més robusta per a diferents casos de mida de tensor i de ràfega.

La metodologia proposada obté molts bons resultats en el cas del senyal del sensor de nivell estudiat. Fins i tot supera molt clarament una de les metodologies tensorials que es fan servir per a restaurar dades, l'algoritme CP-Wopt, amb el que s'han contrastat resultats simulant les mateixes reconstruccions de ràfegues de mostres perdudes. És provable que el motiu d'aquest fet sigui que l'algoritme presentat està dissenyat específicament per resoldre el problema de les dades perdudes en forma de ràfega, mentre que el CP-Wopt tracta el cas de dades perdudes de forma genèrica.

Com a proposta per continuar avançant es podria seguir en la línia de realitzar varies descomposicions, ja sigui per trobar una configuració òptima genèrica per les dues descomposicions, o modificant el concepte en un algoritme recurrent que provi vàries descomposicions i optimitzi els resultats.

En el mètode tensorial proposat, un cop realitzades les millores que ajuden en el procés de descomposició, seria una bona opció tornar a provar l'efecte de modificar la mida del tensor, per veure si incorporant més dades els resultats se'n veuen beneficiats. Ja que tot i que es va determinar en un experiment inicial que a partir de 7 setmanes de mida de tensor s'estabilitzaven els resultats (figura 5.22), es podria comprovar si les millores proposades han afectat en aquest sentit.

També podria ser interessant buscar una altra manera d'ordenar el tensor, que permeti que la descomposició tregui encara més profit de les dades anteriors i posteriors a la ràfega. Aquesta proposta seria rellevant en cas d'adaptar la metodologia a altres sensors o circumstàncies. En la mateixa línia també seria interessant augmentar l'ordre del tensor passant d'una estructura de tres dimensions a una estructura de quatre. Ja sigui per incorporar nous senyals que proporcionin més informació, o bé per buscar noves interaccions entre dimensions del mateix senyal.

En una altra direcció, una eina que podria ser molt útil de cara a la detecció d'anomalies, seria un predictor capaç d'encertar amb fiabilitat els pròxims valors a mesurar. Amb aquesta eina es podria observar la desviació de la predicció de forma que quan aquesta fos gran, podria indicar un possible error o mal comportament del sensor o del sistema. Es podria adaptar les metodologia proposada per a aquesta finalitat, aprofitant l'àlgebra tensorial per crear una aplicació que ajudaria a monitoritzar l'estat dels dispositius que formen part del sistema de distribució d'aigua.

BIBLIOGRAFÍA

[1] Diego García-Joseba Quevedo Vicenç Puig Santiago Espin Jaume Roquet Cugueró-Escofet, Miquel À. A methodology and a software tool for sensor data validation/reconstruction: Application to the catalonia regional water network. *Control Engineering Practice*, 49(4):159–172, 2016.

[2] Ramon Pérez Gabriela Cembrano Joseba Quevedo Teresa Escobet Vicenç Puig, Carlos Ocampo-Martínez. *Real-Time Monitoring and Operational Control of Drinking-Water Systems*. Springer, Gewerbestrasse 11, 6330 Cham, Switzerland, 2017. URL `https://doi.org/10.1007/978-3-319-50751-4`.

[3] Thomas Kailath, Ali H Sayed, y Babak Hassibi. *Linear estimation*. Prentice Hall, 2000.

[4] PP Vaidyanathan. The theory of linear prediction. *Synthesis lectures on signal processing*, 2(1):1–184, 2007.

[5] Jordi Blanch, Vicenç Puig, Jordi Saludes, y Joseba Quevedo. Arima models for data consistency of flowmeters in water distribution networks. *IFAC Proceedings Volumes*, 42(8):480–485, 2009.

[6] B Lamrini, El-K Lakhal, Marie-Véronique Le Lann, y Louis Wehenkel. Data validation and missing data reconstruction using self-organizing map for water treatment. *Neural Computing and Applications*, 20(4): 575–588, 2011.

[7] Vicenç Puig, Carlos Ocampo-Martinez, Ramon Pérez, Gabriela Cembrano, Joseba Quevedo, y Teresa Escobet. *Real-Time Monitoring and Operational Control of Drinking-Water Systems*. Springer, 2017.

[8] Andrzej Cichocki, Danilo Mandic, Lieven De Lathauwer, Guoxu Zhou, Qibin Zhao, Cesar Caiafa, y Huy Anh Phan. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE Signal Processing Magazine*, 32(2):145–163, 2015.

[9] Tamara G Kolda y Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[10] Marko Filipović y Ante Jukić. Tucker factorizatio with missig data with applicatio to low-rak tesor completio. *Multidimensional systems and signal processing*, 26(3):677–692, 2015.

[11] Richard A Harshman. Foundations of the parafac procedure: Models and conditions for an.^{ex}planatory"multimodal factor analysis. *Working Papers in Phonetics*, UCLA Working Papers in Phonetics, 16:1–84, 1970.

[12] Tatsuya Yokota, Qibin Zhao, y Andrzej Cichocki. Smooth parafac decomposition for tensor completion. *IEEE Transactions on Signal Processing*, 64(20):5423–5436, 2016.

[13] Manoj Gupta Ashok Kumar Nagawat Suresh Kumar, Papendra Kumar. Performance comparison of median and wiener filter in image de-noising. *International Journal of Computer Applications*, 12(4):27–31, 2010.

[14] Louis L. Scharf J. Scott Goldstein, Irving S. Reed. A multistage representation of the wiener filter based on orthogonal projections. *IEEE Transactions on information theory*, 44(7):2943–2959, 1998.

[15] Yiteng (Arden) Huang Simon Doclo Jingdong Chen, Jacob Benesty. A multistage representation of the wiener filter based on orthogonal projections. *IEEE Transactions on audio, speech and language processing*, 14 (4):1218–1234, 2006.

[16] MARAPAREDDY. R. Restoration of burred images using wiener filtering. *International Journal Of Electrical, Electronics And Data Communication*, 5(8):45–49, 2017.

[17] Lin Chin-Wei Chen B. Multiscale wiener filter for the restoration of fractal signals: wavelet filter bank approach. *Signal Processing, IEEE Transactions on*, 42:2972–2982, 1994.

[18] Nicholas D Sidiropoulos, Lieven De Lathauwer, Xiao Fu, Kejun Huang, Evangelos E Papalexakis, y Christos Faloutsos. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing*, 65(13):3551–3582, 2017.

[19] S. Espin J. Roquet J. Quevedo, J. Pascual. Data validation and reconstruction for performance enhancement and maintenance of water networks. *IFAC-PapersOnLine*, 49(28):203–207, 2016.

[20] Karina Gibert, Miquel Sànchez-Marrè, y Joaquín Izquierdo. A survey on pre-processing techniques: Relevant issues in the context of environmental data mining. *AI Communications*, 29(6):627–663, 2016.

[21] Karina Gibert. Mixed intelligent-multivariate missing imputation. *International Journal of Computer Mathematics*, 91(1):85–96, 2014.

[22] Chao-Ying Joanne Peng Yiran Dong. Principled missing data methods for researchers. *SpringerPlus*, 2(222):1–17, 2013.

[23] JL Schafer. *Analysis of incomplete multivariate data*. Chapman and Hall/CRC, London, 1997. URL `ISBN9780412040610-CAT#C4061`.

[24] Xiao-Li Meng John Barnard. Applications of multiple imputation in medical studies: from aids to nhanes. *Statistical Methods in Medical Research*, 8:17–36, 1999.

[25] François Husson Julie Josse. Handling missing values in exploratory multivariate data analysis methods. *Journal de la Société Française de Statistique*, 1253(2):79–99, 2012.

[26] Kam CM Collins LM, Schafer JL. A comparison of inclusive and restrictive strategies in modern missing data. *Psychological Methods*, 6(4):330–351, 2001.

[27] Daniel M. Dunlavy, Tamara G. Kolda, y Evrim Acar. Poblano v1.0: A matlab toolbox for gradient-based optimization. Technical Report SAND2010-1422, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, march 2010.

[28] Brett W Bader, Tamara G Kolda, et al. Matlab tensor toolbox version 2.6, available online, february 2015. *URL http://www. sandia. gov/˜ tgkolda/TensorToolbox/index-2.6. html*, 2015.

[29] M-V. Le Lann L Wehenkel Lamrini B., El-K. Lakhal. Data validation and missing data reconstruction using self-organizing map for water treatment. *Neural Computing and Applications*, 20(4):575–588, 2011.

[30] Saeed V Vaseghi. *Advanced digital signal processing and noise reduction*. John Wiley & Sons, 2008.

[31] George EP Box y Gwilym M Jenkins. *Time series analysis: forecasting and control, revised ed*. Holden-Day, 1976.

[32] SK Mitter. Linear estimation-t. kailath, ah sayed, and b. hassibi. *IEEE TRANSACTIONS ON AUTOMATIC CONTROL AC*, 48(1):177–182, 2003.

[33] Marco F Duarte y Richard G Baraniuk. Kronecker compressive sensing. *IEEE Transactions on Image Processing*, 21(2):494–504, 2012.

[34] Zidong Wang, Fuwen Yang, Daniel WC Ho, y Xiaohui Liu. Robust finite-horizon filtering for stochastic systems with missing measurements. *IEEE Signal Processing Letters*, 12(6):437–440, 2005.

[35] Jeffrey Humpherys, Preston Redd, y Jeremy West. A fresh look at the kalman filter. *SIAM review*, 54(4):801–823, 2012.

[36] Heikki Junninen, Harri Niska, Kari Tuppurainen, Juhani Ruuskanen, y Mikko Kolehmainen. Methods for imputation of missing values in air quality data sets. *Atmospheric Environment*, 38(18):2895–2907, 2004.

[37] María Elisa Quinteros, Siyao Lu, Carola Blazquez, Juan Pablo Cárdenas-R, Ximena Ossa, Juana-María Delgado-Saborit, Roy M Harrison, y Pablo Ruiz-Rudolph. Use of data imputation tools to reconstruct incomplete air quality datasets: A case-study in temuco, chile. *Atmospheric Environment*, 200:40–49, 2019.

[38] Yi Yang, Jianwei Ma, y Stanley Osher. Seismic data reconstruction via matrix completion. *Inverse Problems and Imaging*, 7(4):1379–1392, 2013.

[39] Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[40] Pierre Comon. Tensors: a brief introduction. *IEEE Signal Processing Magazine*, 31(3):44–53, 2014.

[41] Lieven De Lathauwer, Bart De Moor, y Joos Vandewalle. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[42] Morten Mørup. Applications of tensor (multiway array) factorizations and decompositions in data mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):24–40, 2011.

[43] Mikael Sørensen, Lieven De Lathauwer, Pierre Comon, Sylvie Icart, y Luc Deneire. Canonical polyadic decomposition with a columnwise orthonormal factor matrix. *SIAM Journal on Matrix Analysis and Applications*, 33(4):1190–1213, 2012.

[44] Lele Wang, Kun Xie, Thabo Semong, y Huibin Zhou. Missing data recovery based on tensor-cur decomposition. *IEEE Access*, 6:532–544, 2018.

[45] Tamara Gibson Kolda. Multilinear operators for higher-order decompositions. Technical report, Sandia National Laboratories, 2006.

[46] Evrim Acar, Daniel M Dunlavy, Tamara G Kolda, y Morten Mørup. Scalable tensor factorizations for incomplete data. *Chemometrics and Intelligent Laboratory Systems*, 106(1):41–56, 2011.

[47] Zemin Zhang, Gregory Ely, Shuchin Aeron, Ning Hao, y Misha Kilmer. Novel methods for multilinear data completion and de-noising based on tensor-svd. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 3842–3849, 2014.

[48] Zemin Zhang y Shuchin Aeron. Exact tensor completion using t-svd. *IEEE Trans. Signal Processing*, 65(6):1511–1526, 2017.

[49] Daniel Kressner, Michael Steinlechner, y Bart Vandereycken. Low-rank tensor completion by riemannian optimization. *BIT Numerical Mathematics*, 54(2):447–468, 2014.

[50] Jordi Sole-Casals, Cesar F Caiafa, Qibin Zhao, y Adrzej Cichocki. Brain-computer interface with corrupted eeg data: A tensor completion approach. *Cognitive Compututation*, 10:1062, 2018.

[51] Jakub Langhammer y Julius Česák. Applicability of a nu-support vector regression model for the completion of missing data in hydrological time series. *Water*, 8(12), 2016. ISSN 2073-4441. doi: 10.3390/w8120560.

[52] Michael Ahlheim, Oliver Frör, Jing Luo, Sonna Pelz, y Tong Jiang. Towards a comprehensive valuation of water management projects when data availability is incomplete — the use of benefit transfer techniques. *Water*, 7(5):2472–2493, 2015. ISSN 2073-4441. doi: 10.3390/w7052472.

[53] Iguniwari Thomas Ekeu-wei, George Alan Blackburn, y Philip Pedruco. Infilling missing data in hydrology: Solutions using satellite radar altimetry and multiple imputation for data-sparse regions. *Water*, 10(10), 2018. ISSN 2073-4441. doi: 10.3390/w10101483.

[54] Qun Zhao, Yuelong Zhu, Dingsheng Wan, Yufeng Yu, y Xifeng Cheng. Research on the data-driven quality control method of hydrological time series data. *Water*, 10(12), 2018. ISSN 2073-4441. doi: 10.3390/w10121712.

Apèndix A

LINEAR PREDICTION TECHNIQUES FOR PERFORMANCE ENHANCEMENT AND MAINTENANCE OF WATER NETWORKS USING SCADA DATA

iEMSs International Congress on Environmental Modelling and Software

9th International Congress on Environmental Modelling and Software
Fort Collins, Colorado, USA, Mazdak Arabi, Olaf David, Jack Carlson, Daniel P. Ames (Eds.)
https://scholarsarchive.byu.edu/iemssconference/2018/

Linear prediction techniques for performance enhancement and maintenance of water networks using SCADA data

Arnau Martí[a,b], Pere Marti-Puig[a], Moises Serra-Serra[a], Manuel Pardo Méndez[b]

[a] *Data and Signal Processing Group, U Science Tech, University of Vic - Central University of Catalonia, c/ de la Laura 13, 08500 Vic, Catalonia, Spain (arnau.mati,pere.marti,moises.serra@uvic.cat)*

[b] *AVSA Aigües Vic, Carrer de la Riera, 6, 08500 Vic, Catalonia, Spain(amarti,mpardo@aiguesvic.com)*

Abstract: From the analysis of the data captured in real time through a SCADA system, a contribution to improving the management of drinking water distribution and the early detection of anomalies is presented. In a real water network, the SCADA system must periodically acquire, store and validate the data collected by sensor measurements to achieve accurate network monitoring. For each sensor measurement, the raw data is usually represented by one-dimensional time series which must be validated before further use to ensure the reliability of the results obtained. In the present approach, we use linear predictors to verify data, detect outliers and restore missing values as well as to forecast different variables at different time intervals. The comparison of the predictions with measurements also serves to generate an error which is reported to an expert through warnings when it is unusually high. This human operator tries to associate significant prediction errors with pump configuration changes or system failures. The work mainly focuses on the predictor's configuration at different temporal levels.

Keywords: water network, SCADA system, Linear prediction, condition-monitoring.

1 INTRODUCTION

In condition monitoring systems working on data collected by SCADA systems, it is critical to verify the integrity of the data received. Otherwise, errors are introduced in the very first step of the processing chain to propagate all along. The presented work is done in the Potable Water Treatment Station (PWTS) of Aigües de Vic S.A. where there are two basic problems with the sensors received information. Sometimes samples of the data series measurements are lost or suddenly false data are received. The first case usually may be because of a sensor failure. The second is caused by errors in the communications system that transmits and stores the information in the central database. To solve or minimize the effect of these troubles, the data of several PWTS sensors, stored by the SCADA system, have been analysed. As indicated by some investigations (Lamrini et al. 2011) it is possible to verify the integrity of the data and validate the obtained samples using predictive techniques. With these tools, even, we can reconstruct discarded samples.

The steps to make the analysis of the data will be the following: collect the historical signal data, make a validation of each sample according to basic criteria, perform a prediction of each sample to discard the incoherent samples and finally reconstruct the discarded samples (Quevedo et al. 2016; Blanch et al. 2009). The method presented and tested to predict the signal is based on a Wiener FIR (Finit Impulse Response) filter, which uses the previous data samples to make the estimation of the next sample. The difficult will be to get a filter configuration that allows estimating the lost or discarded data with reliability. Some modifications will be made to improve the initial algorithms and refine the predictions.

2. MATERIALS AND METHODS

2.1 Data Acquisition

The data history provided by SCADA d'Aigües de Vic S.A. will be used. All signal sensors contained in the ETAB remains in a SQL database in the Aigües de Vic S.A. server, where information has been stored since the beginning of 2014. There are mainly three types of signals. Those provided by flow meters installed in the pipes of the PWTS, which can be data of instant flow or volume of accumulated water. And those supplied by the level sensors of the water tanks. The study initially focuses on the level of water containers. The SCADA that stores the information is centralized on a computer which causes loss of data when this computer fails or is not available. The SCADA combines different communication systems to transmit the data from the sensors to the central computer. In some cases, it is necessary to use repeaters via radio that also causes data loss often. The data is collected every 5 minutes, except for some remote sensors that may have a minor frequency.

2.2 Software used

The Matlab program is used to manipulate and process all the information. This program allows data processing using scripts and includes a toolbox to connect to SQL databases so that all the database information can be easily captured and transferred to vectors with which the Matlab performs any mathematical treatment.

2.3 Data Preprocessing

Before processing the data, the system must ensure that certain integrity conditions are met (Cugueró-Escofet et al. 2016; Quevedo Casín et al. 2017). For this reason, the data vectors and associated dates are checked by validating each of the samples according to their physical and temporal characteristics. The following items are checked. The data read must satisfy the periodicity condition set by the SCADA system that stores the information. This means that the samples must have a temporal separation of 5 minutes and synchronized. For instance, that the minutes must always be multiples of 5 and the seconds of 0. When an inconsistent time is found we decided to eliminate the sample as, it has been observed that, in most cases, the value is also inconsistent. NaNs (Not a Number) are set in temporary positions where no data is available. The units of each magnitude are checked and correct if proceed. The tank level signals collected in the PWTS are done as a percentage and therefore can only take values from 0 to 100. It is checked that the increments of the magnitudes do not exceed the physical restrictions (the level in the tanks cannot vary more than 1-5% in 5 minutes).

2.4 The Wiener predictors

To predict lost samples, we propose a linear estimator that is implement with a finite impulse response filter (FIR). Linear estimators correspond to linear filter structures that are designed following a statistical criterion of minimizing the estimation error, in our case the Wiener filtering technique, which obtain the filter coefficients by minimizing the cost function $E[|e(n)|^2]$, where e(n) is the estimation error, $E[\cdot]$ denotes the expectation operator, and and $|\cdot|$ the Euclidian norm. Figure x, shows a diagram and its development for the case that applies. So, we have:

$$e_n = x_n - \hat{x}_n = x_n - \mathbf{a}^T \boldsymbol{x} \tag{1}$$

Taking into account that vector **a** contains the filter coefficients and vector $\boldsymbol{x}$ the samples used to predict $\hat{x}_n$, as follows:

$$\mathbf{a}^T = [a_0 \quad \cdots \quad a_{L-1}]\ ; \quad \boldsymbol{x}^T = [x_{n-1} \quad \cdots \quad x_{n-L}] \tag{2}$$

The Wiener filter is designed so as to minimize the mean square error (MMSE) criteria stated as:

$$a_l = \arg min\, E[|e_n|^2] \tag{3}$$

Using vector notation the FIR derivation to predict $\hat{x}_n$is quite straightforward. Let's see:

$$E[|e_n|^2] = E[e_n e_n{}^T] = E[(x_n - \mathbf{a}^T\boldsymbol{x})(x_n - \boldsymbol{x}^T\mathbf{a})] = E[x_n x_n - x_n\boldsymbol{x}^T\mathbf{a} - \mathbf{a}^T\boldsymbol{x}x_n + \mathbf{a}^T\boldsymbol{x}\boldsymbol{x}^T\mathbf{a}] \tag{4}$$

$$E[|e_n|^2] = E[x_n x_n] - 2E[x_n \boldsymbol{x}^T]\mathbf{a} + \mathbf{a}^T E[\boldsymbol{x}\boldsymbol{x}^T]\mathbf{a} \quad (5)$$

Being a quadratic function, the resolution of the equation will provide us the minimum of the function

$$\frac{\partial E[|e_n|^2]}{\partial \mathbf{a}} = -E[x_n \boldsymbol{x}^T] + E[\boldsymbol{x}\boldsymbol{x}^T]\mathbf{a} = 0 \quad (6)$$

which is:

$$\mathbf{a} = E[\boldsymbol{x}\boldsymbol{x}^T]^{-1} E[x_n \boldsymbol{x}^T] = \boldsymbol{R}_{\boldsymbol{xx}}{}^{-1} \boldsymbol{r}_{\boldsymbol{xx}} \quad (7)$$

Note than for real magnitudes $\boldsymbol{R}_{\boldsymbol{xx}}$ is a symmetric Toeplitz matrix:

$$\boldsymbol{R}_{\boldsymbol{xx}} = E[\boldsymbol{x}\boldsymbol{x}^T] = \begin{bmatrix} r_0 & \cdots & r_{L-1} \\ \vdots & \ddots & \vdots \\ r_{L-1} & \cdots & r_0 \end{bmatrix} \qquad \boldsymbol{r}_{\boldsymbol{xx}} = E[x_n \boldsymbol{x}^T] = \begin{bmatrix} r_1 \\ \cdots \\ r_L \end{bmatrix} \quad (8)$$

A sliding time window is used to estimate the autocorrelation values $(r_0, \cdots, r_L)$. The window and the filter sizes are selected after a set of experiments.

2.5 Proposed method for signal reconstruction

It is clear that predictions are getting worse and worse as they try to anticipate more time in the future. This is a problem when bursts of values are lost. To minimize errors and reconstruct the databases, we propose to make a prediction called forward $\hat{f}_i$ to fill the lost values from the start of the burst and a backward prediction, $\hat{b}_i$ that supplies the missing values from the end of the burst to the past. As the values are more reliable at the extremes, to fill a burst of *N* missing samples (i goes from 1 to N) we weigh the estimate as follows:

$$\hat{x}_i = \frac{\hat{f}_i(N-i) + \hat{b}_i i}{N} \quad (9)$$

3. RESULTS

Two experiments have been designed to find the filter length and the optimal number of samples necessary for calculating the correlation. The first one computes the MSE for various values of the order of the filter, to be able to see the effect of each one of the parameters. It was determined that from an order of 245 coefficients there is practically no improvement in the MSE. Regarding the delay, as it was to foresee, more delay more error, in an entirely straightway.To be able to choose the number of samples required to obtain good estimations of the correlation, since they are not stationary systems, a test is performed in which the MSE is calculated for several values of the number of samples used. In this experiment, it is determined that around 250 samples are the most indicated. Some samples were removed from the original data to simulate intervals of lost data to verify de performance of the method. The graphic shows the result of the three proposed method, Normal FIR, Reverse FIR and mix FIR.

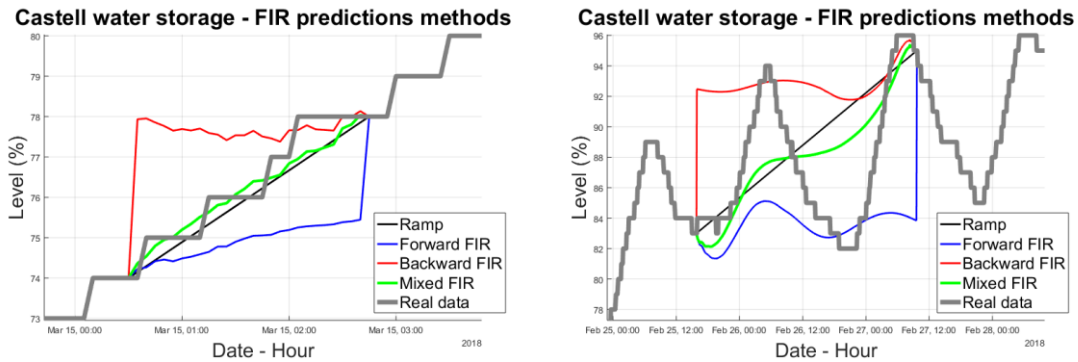

Figure 1 Two examples of simulating a burst of sample loss and reconstruction. The results are compared with the true data

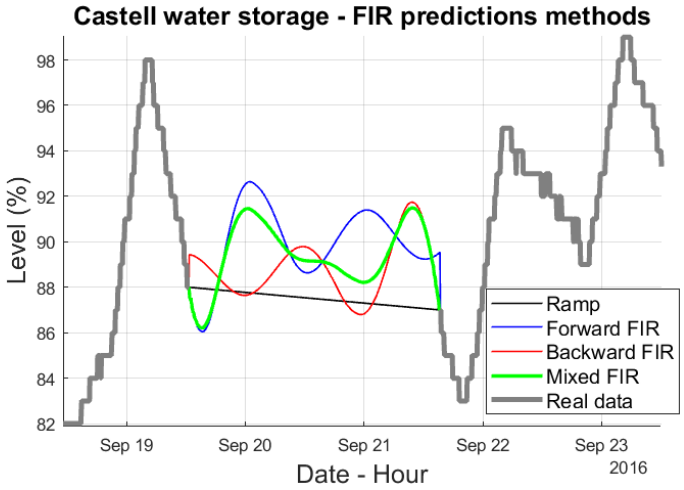

Figure 2 Reconstruction of a lost data stream.

1.5 CONCLUSIONS

The normal method with the FIR filter works correctly in the case of few consecutive lost samples. But in this case, a simple ramp or line joining the last data, before the first lost data, and the next data received after the last lost data, is a good approximation too. When the lost data burst is large (more than 25 samples, which represents some hours) the FIR method tends to get away from the reality. The filter configuration is not simple, because of the order and the number of samples for the autocorrelation calculation is critical and depends on the number of consecutive sample reconstructions that it's necessary to estimate. If the configuration is not sufficiently accurate, the FIR filter prediction error increases the longer is the burst of lost data. In this case, the proposed mix method between normal FIR and reverse FIR minimizes the prediction error. In addition, this method maintains the signal coherence, even if it does not exactly estimate the data burst.

ACKNOWLEDGMENTS

This work is partially supported by the Catalan Government under grant 2015DI40. Just thank the company Aigües de Vic S.A. that allowed us access to their databases to perform this research.

REFERENCES

Blanch, Jordi, Vicenç Puig, Jordi Saludes, and Joseba Quevedo. 2009. ARIMA Models for Data Consistency of Flowmeters in Water Distribution Networks. IFAC Proceedings Volumes 42 (8). Elsevier: 480–85. https://doi.org/10.3182/20090630-4-ES-2003.00080.

Cugueró-Escofet, Miquel À., Diego García, Joseba Quevedo, Vicenç Puig, Santiago Espin, and Jaume Roquet. 2016. A Methodology and a Software Tool for Sensor Data Validation/reconstruction: Application to the Catalonia Regional Water Network. Control Engineering Practice 49 (April): 159–72. https://doi.org/10.1016/j.conengprac.2015.11.005.

Lamrini, B., El-K. Lakhal, M-V. Le Lann, and L Wehenkel. 2011. Data Validation and Missing Data Reconstruction Using Self-Organizing Map for Water Treatment. Neural Computing and Applications 20 (4): 575–88. https://doi.org/10.1007/s00521-011-0526-5.

Quevedo, J., J. Pascual, S. Espin, and J. Roquet. 2016. Data Validation and Reconstruction for Performance Enhancement and Maintenance of Water Networks. IFAC-PapersOnLine 49 (28). Elsevier: 203–7. https://doi.org/10.1016/J.IFACOL.2016.11.035.

Quevedo Casín, Joseba Jokin, Diego García Valverde, Vicenç Puig Cayuela, Jordi Saludes Closa, Miquel Àngel Cugueró, Santiago Espin Basany, Jaume Roquet, and Fernando Valero Cervera. 2017. Real-Time Monitoring and Operational Control of Drinking-Water Systems. Edited by Vicenç Puig, Carlos Ocampo-Martínez, Ramon Pérez, Gabriela Cembrano, Joseba Quevedo, and Teresa Escobet. Advances in Industrial Control. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-50751-4.

Apèndix B

DIFFERENT APPROACHES TO SCADA DATA COMPLETION IN WATER NETWORKS

water

MDPI

Article

Different Approaches to SCADA Data Completion in Water Networks

Pere Marti-Puig [1,*,†], Arnau Martí-Sarri [1,2,†] and Moisès Serra-Serra [3]

[1] Data and Signal Processing Group, U Science Tech, University of Vic—Central University of Catalonia, c/de la Laura 13, 08500 Vic, Catalonia, Spain; arnau.marti@uvic.cat
[2] Aigues de Vic S.A., c/Santiago Ramon y Cajal 60, 08500 Vic, Catalonia, Spain
[3] MECAMAT Group, U Science Tech, University of Vic—Central University of Catalonia, c/de la Laura 13, 08500 Vic, Catalonia, Spain; moises.serra@uvic.cat
* Correspondence: pere.marti@uvic.cat; Tel.: +34-93-881-55-19
† These authors contributed equally to this work.

Received: 19 March 2019; Accepted: 9 May 2019; Published: 16 May 2019

Abstract: This work contributes to the techniques used for SCADA (Supervisory Control and Data Acquisition) system data completion in databases containing historical water sensor signals from a water supplier company. Our approach addresses the data restoration problem in two stages. In the first stage, we treat one-dimensional signals by estimating missing data through the combination of two linear predictor filters, one working forwards and one backwards. In the second stage, the data are tensorized to take advantage of the underlying structures at five minute, one day, and one week intervals. Subsequently, a low-range approximation of the tensor is constructed to correct the first stage of the data restoration. This technique requires an offset compensation to guarantee the continuity of the signal at the two ends of the burst. To check the effectiveness of the proposed method, we performed statistical tests by deleting bursts of known sizes in a complete tensor and contrasting different strategies in terms of their performance. For the type of data used, the results show that the proposed data completion approach outperforms other methods, the difference becoming more evident as the size of the bursts of missing data grows.

Keywords: water networks; SCADA data; tensor completion; tensor decomposition

1. Introduction

In a real distribution network of drinking water, the SCADA (Supervisory Control and Data Acquisition) system must periodically acquire, store, and validate the data collected by sensors to achieve accurate monitoring and control of the system. However, before these data can be used to improve the management of the system and the early detection of anomalies, their integrity must be verified. Otherwise, the models appear distorted and errors propagate from the very beginning. Therefore, the raw data coming from each sensor measurement, usually represented by a one-dimensional time series, must be validated before further use, as it is the only way to ensure the reliability of the results obtained afterwards. One of the most significant problems in data management is the losses that occur when either sensors or communication links fail. In both cases, this results in the loss of a burst of samples. The problem of managing lost data is ubiquitous in many situations and is especially challenging when it manifests itself in long bursts. Dealing with incomplete data is very common in real-life cases, and it is usual to find such cases in water [1] and hydrological data management [2–4]. The current work used data from the Drinking Water Treatment Station (DWTS) of Aigües de Vic S.A. For this purpose, data from several DWTS sensors that are stored by the SCADA system in a database were analyzed. The main problem with the usage of

these data for model generation is the amount of lost data that appears in the form of bursts. All of these bursts of missing data have a maximum length of <24 h that depends on the company's fault repair protocols. In Aigues de Vic S.A., there are five technicians available during working hours (4:00–00:00), with at least one of them present in the DWTS at any moment. In the water network, there are six operators working from 8:00 to 18:00 with a 2 h break for lunch. To cover the supervision of the DWTS and the water network for 24 h every day, a technician and an operator take turns on duty, even during traditional non-working hours, including the weekends. As indicated by certain research [5], it is possible to verify the integrity of the data and validate the samples obtained by using predictive techniques. Although traditional data completion approaches [5–8] work well when the proportion of missing data is low, their performance is affected when the ratio increases, mainly because these methods rarely take advantage of the hidden structure of the data [9,10]. The capture of the underlying structure of the data can instead be more easily reached by using tensor factorization techniques when data sets have more than two axes of variation [11–14]. When the problem of missing data is considered under the framework of tensors, two main common strategies are contemplated. One is known as *maximization of expectations* and involves the assignment of estimated values to fill in the missing data. The other, marginalization, ignores the missing values (marginalized) during the optimization process [11]. One of the most well known marginalization methods is the CP-Wopt (CP Weighted OPTimization), which was first introduced in [9]. This algorithm was derived from the CANDECOMP/PARAFAC (CP) tensor factorization for a case with missing data, in which it was reformulated as a weighted least-squares problem that only takes the known entries into account. In comparison with the existing tensor completion methods, our approach is based on an ad-hoc strategy that has proven to be effective when errors occur in bursts. It is based on two main steps: In the first one, a low dimensional method, such as a linear interpolation or a Wiener predictor, is employed to fill in the missing values, while, in the second step, the data are *tensorized* according to their naturally observed periodicity so that the structure of the data can be exploited beyond a single dimension, and therefore, their common features can be captured in five minute, daily, and weekly intervals. In terms of validation of the results, when vectorizing the final output, it has been observed that reconstructions of tensors from low-range models collect the daily oscillations observed in the data better than low-dimensional techniques; this phenomenon becomes increasingly evident as the burst gets longer. However, when assigning the data burst retrieved to the lost positions in order to fill the gap, discontinuities within the original data are observed at the extremes. The ad-hoc observation is such that an offset correction in the recovered data highly improves the reconstruction. This correction is based on the way that the forwards and backwards Wiener predictors are combined in the first stage of the method.

Experiments where bursts of a given size are randomly erased have shown a performance improvement with respect to the traditional methods, including the CP-Wopt [9]. The current study proposes a new algorithm that uses a fixed low-order tensor decomposition which is computationally efficient.

The algorithm encompasses three strategies in the first part of the algorithm. The first two, which are briefly explained in Section 2, are very basic, but were considered as a result of their use by Aigües de Vic S.A. The third consists uses prediction techniques, such as FIR (Finite Impulse Response) filters from Wiener [15]. The design of the predictors has two key elements: the order of the filter and the method used to estimate the auto-correlation of the signal, which is necessary for the calculation of the filter coefficients [15].

On the other hand, tensor techniques provide an effective representation of the structural properties of multidimensional data. Some of the most powerful tools of the tensor algebra are the decompositions that allow the discovery of the interrelations between dimensions. There are various decompositions, but the CANDECOMP/PARAFAC (CP) [16,17] and the Tucker [18–20] are the most well known and widely used and therefore, were considered in the present study. There is an abundance of literature on tensors. Most of the tutorials available introduce the topic from

the perspective of their applications in the fields of machine learning and/or signal processing. Some quality references that are very useful for a rigorous first contact with the subject are [12–14,21]. One of the many uses of tensor algebra is in the field of missing data recovery or tensor completion, in which it has been successfully applied. For instance, in [22], it was used to recover missing values in the visual data; in [23,24], the proposed method was mainly applied to the recovery of traffic data values; and in [25], a low-n-rank tensor recovery method was introduced. Furthermore, in [26], a completion tensor method, which automatically determines the rank of the decomposition, was put forward. However, the tensor completion strategy proposed in [9] is one of the most developed methods, and its corresponding algorithm can be used from open access libraries [27,28].

In a nutshell, when data loss is spread uniformly or even in short bursts (less than 30–40 samples, for our type of signals), all of the methods perform similarly in practice. Above that sample length, the performance of the commonly used methods begins to decrease. However, in practice, data are generally lost in bursts, which can also be very long. So, our proposal was motivated by the need to improve the performance of currently used data completion methods. In this work, we provide evidence of the improvement achieved in comparison with more straightforward and faster methods at the expense of increasing the complexity of the algorithms slightly.

After the introduction, this work continues with Section 2, where the details for reproducing the results are explained. Within this, aspects related to the data and their pre-processing are presented. The section also covers the three "non-tensor" methods used to impute values and a tensor introduction explaining how data are "tensorized", as well as some of the main tensor properties and the two most well known low-rank decompositions. To compare algorithms, a statistical strategy is put forward which consists of erasing known data (in bursts) inside the tensor in order to take an objective measure after comparing the eliminated real values with the recovered ones. The CP-Wopt tensor completion method is tested in the context of our problem and, based on the obtained results, a solution is proposed involving a final step called the "offset continuity correction". Section 3 presents a set of experiments that were designed to evaluate a range of factors, such as the configuration of the FIR, the optimal size of tensor decompositions for both Tucker and CP models, the optimal size of the tensor, and the evaluation of the algorithms according to the length of the lost burst. The main points are summarized in Section 4, and examples are included to illustrate how our approach works in terms of the representation of the recovered data.

2. Materials and Methods

2.1. Data Acquisition

The data used for this research were provided by Aigües de Vic S.A., who have a SCADA system that manages the information collected by the sensors of the DWTS and the water distribution network. The network of pipes is 428.185 km in length, and there are 2800 mechanical water meters, which need to be read by an operator every three months, as well as 21,000 remote water meters, which are connected to a database and send a daily reading. The SCADA receives 1309 different signals from the DTWS and the network, such as instant flow or accumulated water volume from a flowmeter; the pressure of a manometer; the fullness percentage of a water deposit from a level sensor; the value assigned to a pump frequency converter; pH meter data; and on/off signals of valves, pumps, and water deposit buoys.

Their measurements are stored every five minutes in a Structured Query Language (SQL) database of the SCADA server, which has been in operation since 2014, with the majority of signals saved from October 2015 onwards. The level sensor signal studied in this paper has been collected since 1 October 2015 at 07:50. However, an important loss of data occurred from 8 September 2017 at 11:15 until 27 August 2018 at 22:30. The maximum number of weeks configured in the tensor tests is 31. Because of this, approximately 8 months of useful data were required for this experiment. The tested week was chosen for two reasons. Firstly, it had only a small amount of lost original data, which allowed

verification to occur when we restored the deliberately deleted data. There were three weeks, between 23 January 2017 and 12 February 2017, with only three empty positions in the original tensor, which we restored with linear methods. Secondly, good data were obtained for the 14 weeks prior to a following the chosen week, thus ensuring the reliability of the FIR with the highest order and largest tensor calculations. In the selected weeks, the biggest range of used data was from 17 October 2016 to 1 May 2017.

The server that manages all of the information is centralized in a computer. Occasionally, this computer fails or its communication with the sensors is interrupted, resulting in the loss of data. Aigües de Vic S.A. allows access to the data history of any sensor, and we obtained this information using Matlab®from MathWorks (Natick, MA, USA). We started with the Matlab 2016a version; however, during the study we updated it to the Matlab 2018b version. This program is able to execute SQL scripts, making it easy to communicate with the database directly and thus obtain all the required data. This study focused on the sensor that measures the water deposit level of *Castell d'en Planes*, which supplies water to the city of Vic, the capital of Osona region located in the north-east of Catalonia. The population of Vic is 48,287 people (in 2016) and it has an area of 30.6 Km^2. It is very important that this deposit is never fully emptied, because it would leave the city of Vic without any water.

2.2. Data Pre-Processing

The first step that had to be taken before the sensor measurements of the water level could be obtained was to test the integrity of these measurements and to discard the incoherent data [7,8] that do not follow the laws of physics or the particular rules of the sensor environment (for instance, we know that a level sensor measurement of 0 is incorrect because this deposit never has been empty.) The NaNs (Not a Number) classification was given to positions where no data are available, either because they were not saved or because they had been discarded. The physical and temporal characteristics of every collected sample had to be verified. In the temporal domain, two checks were made. The first one was related to synchronization. The read data had to satisfy the periodicity condition set by the SCADA server, which stores the samples every 5 min. This means that the registered minutes digit of every sample must always be a multiple of 5, and the second digit must be a 0. A decision was made to eliminate samples with an inconsistent time, as hown in Figure 1, because, in most cases, the value of this sample was also inconsistent (in the case of the targeted level sensor, 288 of 212,549 samples read as 0 had an incorrect time stamp). The other type of temporal verification made was to ensure that there were no duplicated data. In this case, if the two duplicated values were the same, one was deleted, and if the values were different, the incoherent one (often being 0) was deleted. In the physical domain, the general characteristics of the sensor needed to be considered along with the particular characteristics of the sensor environment. The sensors have minimum and maximum possible values and, in general, certain physical restrictions between the value of a sample and the next one. In the case of the level sensor, the possible values range from 0 to 100 as they are percentages. However, in the case of the *Castell d'en Planes* deposit, which is the water reserve of the city, we can state that the minimum value is 30%. The sensor can only measure values below 30% when the deposit is being cleaned; however, when this happens, the sensor is turned off. Another distinctive detail of this particular deposit is that because of the pumping configuration of water filling and the city water demand, it cannot be emptied or filled at a higher rate than 1% in 5 min. To sum up our case, the range of valid values for the measurements is from 30% to 100%, and the difference between a sample and the next one must be within ±1%. All the samples that do not accomplish these conditions are discounted, as is shown in Figure 2.

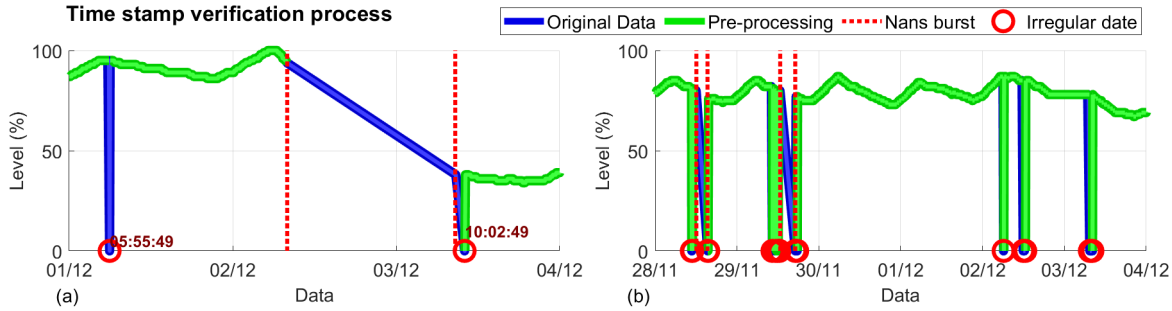

Figure 1. Particular cases of time stamp pre-processing: (**a**) shows a pair of read samples with the irregular date and a burst; (**b**) Shows a pair of bursts and many samples with a value of 0 and irregular date. In both examples, some samples of the pre-processed signal have a value of 0. This is because, at this stage, only the date is pre-processed and there are some samples, read as 0, with a correct time stamp.

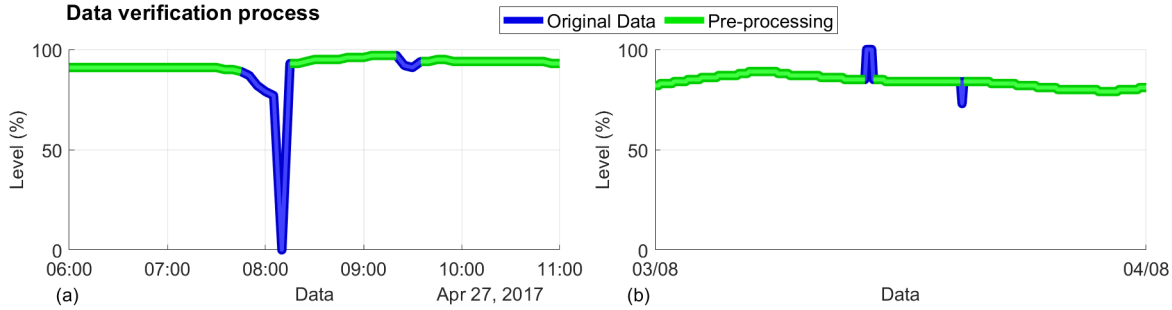

Figure 2. Particular cases of the signal pre-processing step: (**a**) Shows an impossible fall in the level. An operator remembers this case. It was caused by him touching the sensor accidentally. Minutes later he re-positioned the sensor, and approximately 1 h later, a technician re-calibrated the sensor. (**b**) Shows an unusual behavior of the sensor, which spontaneously read incorrect measurements of the level.

2.3. Linear Methods Used for Data Completion

After the pre-processing stage, we were left with a record of samples from the signal level with some gaps due to lost samples (the reasons for this ranged from a SCADA system failure, a communications problem, or a sensor malfunction) and the discarded samples. The most frequent and difficult errors to correct are those caused by bursts, which are often associated with sensor or communication failures. Once data loss is detected, the technical services of the company usually repair the problem and very rarely do the bursts of lost data go on for longer than a day. However, it is common for them to last several hours. Operators have some simple methodologies that they can use to fill the gaps, depending on the data that has been lost and the context in which they were lost. The methods that the operators can use are called the *last sample* method and the *ramp* method. In order to improve the process of filling data, we propose a complementary method, based on the Wiener predictor, which we call the *FIR* method. This third method uses linear prediction techniques, and it is the result of combining forward and backward estimations. When the fault has been repaired, the SCADA system supplies the data again, and because of this, these new data can be used to produce a backwards estimation. We have found that at a certain lost burst size, the latter method usually outperforms the two more simple ones. Figure 3 shows how the above three methods work for two examples. Following this is a brief description of the three methods.

There are different linear techniques for prediction [29] and data imputation [30,31]. Among the most commonly used linear filters, in addition to those of Viener, are techniques using the Kalman filters [32,33], which have applications in air quality [34,35] and seismic [36] data set restoration.

The three methods used are shown below, with emphasis placed the technique based on FIR filtering, which was developed for this particular application.

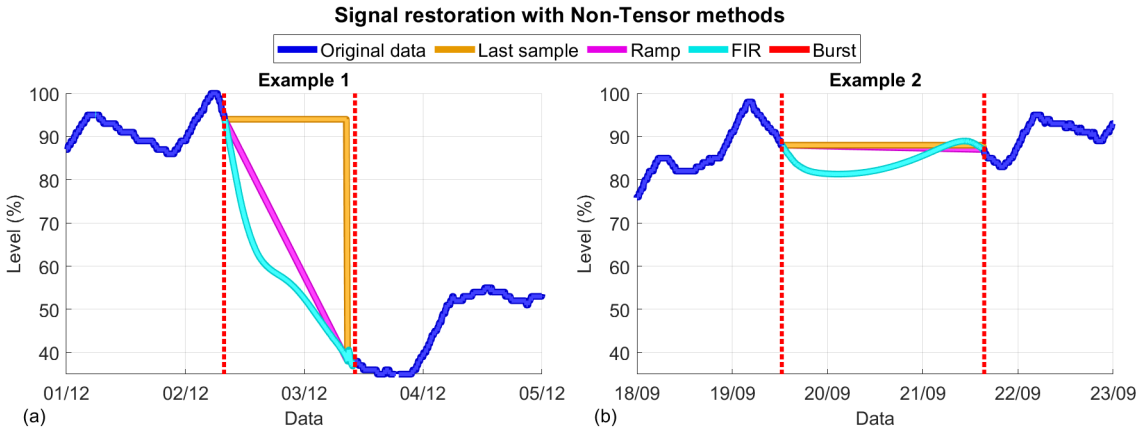

Figure 3. Particular cases of signal restoration using non-tensor methods: (**a**) Shows a middle-length burst where the *last sample* method does not work effectively but the *ramp* and *FIR* (Finite Impulse Response) methods do; (**b**) shows a large-length burst where only the *FIR* method gives a coherent result.

2.3.1. *Last Sample* Method

This is the method used by operators when only the last reliable sample before the burst of lost data is available. Since no extra information is available at the time of restoration, the last reliable value recorded is simply maintained throughout the burst. In other words, all gaps are filled with the same value. Despite its extreme simplicity, this method is adequate if the burst of lost samples is short.

2.3.2. *Ramp* Method

Operators use this other method when they have a reliable pre-burst and post-burst measurement. In this case, a very simple linear approximation is made. The increment/decrement is computed which, when regularly applied to the last known sample, allows the gaps to be filled by linking the last known sample to the next validated sample with a straight line.

2.3.3. *FIR* Method. The Wiener Predictor

A Wiener filter is used to make approximations. It is designed following a statistical criterion in order to obtain the filter coefficients by minimizing the cost function $E[|e_n|^2]$, where e_n is the estimation error, $E[\cdot]$ denotes the expectation operator, and $|\cdot|$ is the Euclidean norm. This error is defined as the difference between sample x_n and its estimation $\hat{x}_n$ which is done by using the FIR filter. According to this, we have $e_n = x_n - \hat{x}_n = x_n - \boldsymbol{a}^T\boldsymbol{x}$ where vector $\boldsymbol{a}^T = [a_0 \dots a_{L-1}]$ contains the filter coefficients and vector $\boldsymbol{x}^T = [x_{n-1} \dots x_{n-L}]$ contains the samples used to make the approximation. We can write the cost function as

$$E[|e_n|^2] = E[e_n e_n^T] = E[(x_n - \boldsymbol{a}^T\boldsymbol{x})(x_n - \boldsymbol{a}^T\boldsymbol{x})^T] = E[x_n x_n^T - x_n\boldsymbol{x}^T\boldsymbol{a} - \boldsymbol{a}^T\boldsymbol{x}^T x_n + \boldsymbol{a}^T\boldsymbol{x}\boldsymbol{x}^T\boldsymbol{a}] \quad (1)$$

which is

$$E[|e_n|^2] = E[x_n x_n] - 2E[x_n\boldsymbol{x}^T]\boldsymbol{a} + \boldsymbol{a}^T E[\boldsymbol{x}\boldsymbol{x}^T]\boldsymbol{a}. \quad (2)$$

The Wiener filter was designed to minimize the mean square error (MSE) criteria. Since Equation (2) is a quadratic function, the solution corresponds with its minimum, which is found after solving

$$\frac{\partial E[|e_n|^2]}{\partial \boldsymbol{a}} = 0, \quad (3)$$

which, when solved, provides the minimum value of the function. As

$$\frac{\partial E[|e_n|^2]}{\partial \boldsymbol{a}} = -E[x_n \boldsymbol{x}^T] + E[\boldsymbol{x}\boldsymbol{x}^T]\boldsymbol{a}, \tag{4}$$

the FIR coefficients will be

$$\boldsymbol{a} = E[\boldsymbol{x}\boldsymbol{x}^T]^{-1} E[x_n \boldsymbol{x}^T] = \boldsymbol{R_{xx}}^{-1} \boldsymbol{r_{xx}} \tag{5}$$

where $\boldsymbol{R_{xx}}$ for real valued magnitudes is a symmetric Toeplitz matrix that takes the form

$$\boldsymbol{R_{xx}} = E[\boldsymbol{x}\boldsymbol{x}^T] = \begin{bmatrix} r_0 & \dots & r_{L-1} \\ \vdots & \ddots & \vdots \\ r_{L-1} & \dots & r_0 \end{bmatrix} \tag{6}$$

and $\boldsymbol{r_{xx}}$ written in terms of the auto-correlation coefficients r_i is $\boldsymbol{r}_{xx}^T = [r_1 \dots r_L]$. Concerning the predictor design, there are two key points: the size of the filter, i.e., the value of coefficient L [15], and the method of estimating the auto-correlation coefficients r_i. A sliding time window of size M is used to achieve this aim according to the expression [37]

$$r_k = \frac{\sum_{i=1}^{M}(x_i - \bar{x})(x_{i-k} - \bar{x})}{\sum_{i=1}^{M}(x_i - \bar{x})^2}, \tag{7}$$

where

$$\bar{x} = \frac{1}{M}\sum_{i=1}^{M} x_i. \tag{8}$$

The window and the filter sizes, M and L, are selected after a set of experiments that are described in the next section. To begin to fill the gap, the first estimation is made. The estimated sample $\boldsymbol{a}^T\boldsymbol{x}$ is used to fill the corresponding position in the burst, and it is also used to update the vector $\boldsymbol{x}$ in order to estimate the next one. Then, the process is repeated until the whole burst has been filled. Although the values of the signal that is to be estimated are highly correlated, as we get further from the last verified sample the estimation error grows. Because of this, the last estimated sample does not correspond with the first verified sample following the end of the burst, causing a loss in the continuity of the signal. We call this estimation *forward* as it goes forwards in time. Similarly, estimation *backward* consists of repeating the same technique, but in this case, using the samples obtained after the burst to make the estimation. The estimation begins at the end of the gap and moves towards the beginning, progressing backwards in time. This estimation behaves in an opposite fashion to the previous one: it works very well with estimations at the end of the gap; however, its performance worsens as its approaches the beginning of the gap. Hence, both estimations are not able to maintain signal continuity throughout the whole gap, but both still correctly capture signal oscillations.

2.3.4. *FIR* Method. Forward and Backward Wiener Predictor Combinations

In order to take advantage of both forward $\hat{f}$ and backward $\hat{b}$ estimations, they are combined in the following way:

$$\text{for} \quad i = 1 \dots N; \qquad \hat{x}_i = \begin{cases} \dfrac{\hat{f}_i + \hat{b}_i}{2} & \text{if } N = 1 \\ \dfrac{(N-i)\hat{f}_i + (i-1)\hat{b}_i}{N-1} & \text{if } N > 1 \end{cases} \tag{9}$$

where i indexes the N lost samples from start to end, following a chronological order. Using this expression, greater weight is placed on the best estimations in each of the extremes, and the continuity

of the recovered signal with the valid data is maintained. Figure 4 shows two examples of data reconstruction using this technique. It can be observed that the forward and backward reconstructions follow the signal trend. Although both maintain the continuity of the signal at one end of the lost data burst, two methods data at the other end. By combining these estimates as proposed, continuity is maintained and a more consistent estimation result is obtained.

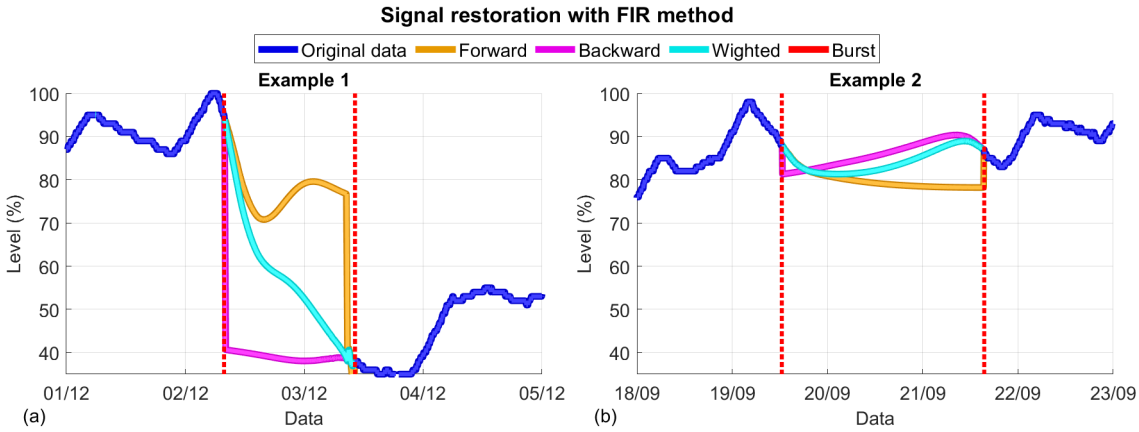

Figure 4. Particular cases of signal restoration with the *FIR* method: (**a**) shows a middle-length burst and (**b**) shows a large-length burst. In both cases, the *forward* and the *backward FIR* methods do not maintain the continuity of the signal but the *Weighted FIR* method does maintain such continuity.

2.4. Tensor-Based Methods for Data Completion

The problem of missing data completion was recently addressed under the tensor framework, and some new strategies appeared. Some of these are derived from well-known decomposition techniques, such as [38,39] for the tensor-Single Value Decomposition (t-SVD), [40] for the Tucker decomposition, and [26,41] for the CP/PARAFAC, while some others strategies originate from less-known methods such as the Riemannian optimization method [42]. As mentioned, tensor methods have been successfully applied in data mining [11] and signal processing [13]. In the presented approach, a second stage/process is added to improve the completion/restoration of the lost samples applied to each of the linear methods, based on the use of tensors, thus forming a new data restoration method that consists of two stages. In the first stage, a linear method is used realize a first estimation of the lost data using some previous and subsequent data around the burst. In the second stage, the data are introduced in a tensor. This tensor is subjected to a simplifying process (decomposition) in a way that allows us to recover the information while avoiding certain inconsistencies generated by the estimation error of the linear methods. This mathematical tool requires that a complete tensor with no missing data is introduced. Because of this, before using the tensors, a previous restoration (imputation) method must be used to fill the gaps caused by lost data.

2.4.1. Definition of a Tensor and Some of Its Basic Properties

A tensor is a container that can arrange data in N-ways or dimensions. The number of dimensions is the order of the tensor. An N-way tensor of real elements is denoted as $\chi \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, and its elements are $x_{i_1,i_2,\ldots,i_N}$. According to this, an $N \times 1$ vector $\mathbf{x}$ is considered to be a first-order tensor, and an $N \times M$ matrix $\mathbf{X}$ is a second-order tensor. A subtensor is a part of the original tensor which is formed by fixing a subset of indexes. It works in the following way:

- By fixing every index but one, the subtensor is a vector, and it is referred to as a *fiber*. For instance, the fibers $\chi_{:,1,1}$, $\chi_{1,:,1}$, and $\chi_{1,1,:}$ are the first column, row, and *tube* of the three-way tensor χ, respectively.

- A matrix, also called a *slice*, is created by fixing all but two tensor indexes. Following the same example, the matrices $\chi_{:,:,k}$, $\chi_{:,j,:}$, and $\chi_{i,:,:}$ of the three-way tensor χ are frontal, vertical, and horizontal slices, respectively.

The process of reshaping tensors into matrices is known as matricization or matrix unfolding and plays an important role in defining the algebraic operations between tensors and matrices. The following points must be taken into account:

- Commonly, the notation $\boldsymbol{X}_{(n)}$ is used to represent the mode-n matricization of $\chi \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, an operation which reshapes χ in a matrix $\boldsymbol{X}_{(n)} \in \mathbb{R}^{I_n \times I_1 I_2 \cdots I_{n-1} I_{n+1} \cdots I_N}$.
- Similarly, the reverse operation of mapping a matrix into a tensor is called unmatricization.

The process of reshaping tensors to vectors is named vectorization of a tensor:

- It is denoted $vec(\chi)$ and reshapes $\chi \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ into a vector $\boldsymbol{x} \in \mathbb{R}^{I_1 I_2 \cdots I_N}$.

Tensor algebra has many similarities but also many surprising differences with matrix algebra. It is particularly important to define the product between matrices and tensors in order to define the type of tensor decomposition to be used. The following two notations are used:

- The mode-n product of a tensor $\chi \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n \times \cdots \times I_N}$ by a matrix $\mathbf{A} \in \mathbb{R}^{J_n \times I_n}$ is denoted as $\zeta = \chi \times_n \mathbf{A}$, with $\zeta \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times J_n \times \cdots \times I_N}$ being the resulting tensor.
- The mode-n product $\zeta = \chi \times_n \mathbf{A}$ also has the following matrix representation $\mathbf{Z}_{(n)} = \mathbf{A}\mathbf{X}_{(n)}$, where $\mathbf{Z}_{(n)}$ and $\mathbf{X}_{(n)}$ are the mode-n matrization of tensors ζ and χ, respectively.

All of these operations and many others are explained in more detail and with graphical examples in [11–13,21]. It is important to note that the reduction of dimensionality and the tensor decomposition techniques allow high-dimensional data to be mapped in a low-dimensional space, conserving the maximum amount of information and making it possible to determine the interactions between dimensions, overcoming the second-order restriction of order two imposed by matrix algebra.

2.4.2. Data Tensorization

The procedure of creating a data tensor from lower-dimensional original data is referred to as *tensorization*. The organization of the data in a container, the tensor, which has more dimensions than the original container allows us to find the relations between dimensions, which are difficult to perceive in more simple structures. In the present case, a three-way tensor of measures is created with the following mode indexes:

- Five minute day intervals: indicates 5 min intervals throughout the day. One day is divided into 288 such intervals.
- Day of the week: indicates the day of the week, from Monday to Sunday, which corresponds to a number from 1 to 7.
- Weeks: indicates n_w, the number of weeks included in the tensor.

The *tensorization* process is shown in Figure 5. Figure 5a shows the evolution of the water levels of a deposit over three weeks. Measures are taken every 5 min and are given as a percentage of the tank capacity as it has regular sections. The staggered effect of the graph shows that the sensors have a limited resolution. The water levels follow specific loading and unloading patterns during the day. There also appears to be some similarity during the same days of the week. For instance, the behavior at the weekends appears to differ from that observed during working days, and because of this, a weekly pattern may well exist. In order to explore and take advantage of possible regularities in the data structure, the data are packed in a tensor χ of size $288 \times 7 \times n_w$. The first dimension corresponds to the number of measures taken during the daytime cycle, the second dimension corresponds to the

days of the week, and the third, n_w, refers to the number of weeks considered. Therefore, Figure 5a shows a representation of the packing of three weeks of data according to the explained criterion to finish building a tensor $\chi^{288\times7\times3}$ of size $288 \times 7 \times 3$. Within that, Figure 5b represents the data corresponding to the first week $\chi(:,:,1)$, while Figure 5c,d portray the data corresponding to the second and third weeks, respectively. According to this notation, $\chi(:,:,1)$, $\chi(:,:,2)$, and $\chi(:,:,3)$ are matrices, and $\chi(:,1,1)$ is a column vector that contains measurements made on Monday of week 1. From the $\chi^{288\times7\times3}$ tensor, the original signal can be retrieved from the operation $vec(\boldsymbol{X}_{(1)})$, $\boldsymbol{X}_{(1)}$, which is the mode-1 matricization of χ.

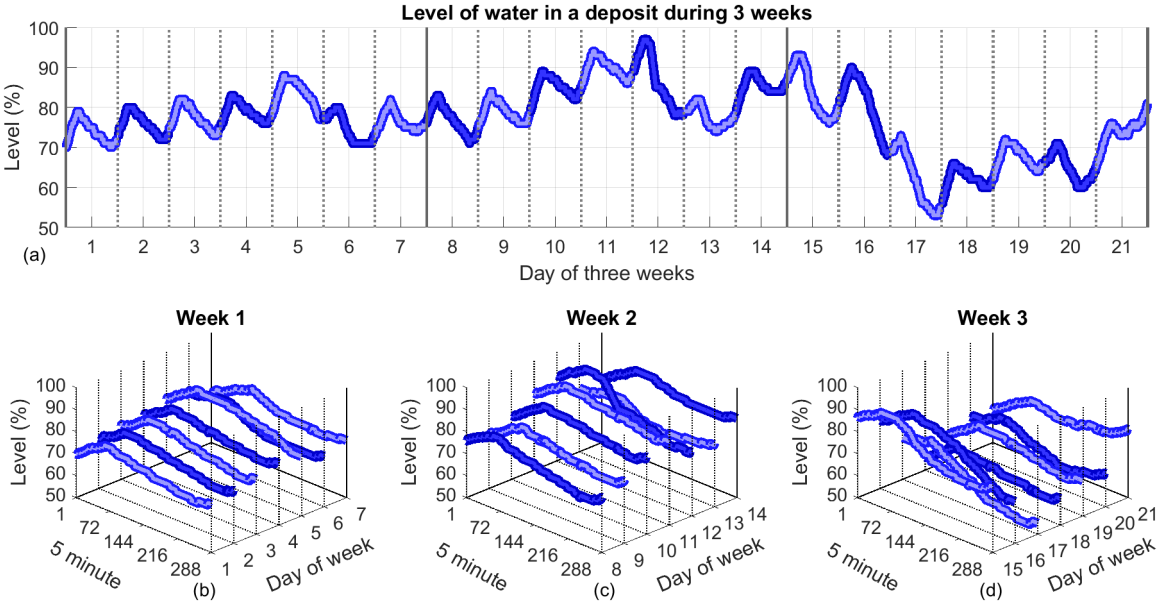

Figure 5. Representation of vector folding **x**, corresponding to the measurements of three weeks in a tensor $\chi^{288\times7\times3}$: (**a**) represents the signal **x** where the fine vertical lines denote the daily separation and the bold vertical lines show the weekly separation; (**b**) represents the information given in $\chi(:,:,1)$ of week 1, in which each vector $\chi(:,i,1)$, $i \in [1,7]$ provides the daily measures of day i; (**c**) shows the same representation for $\chi(:,:,2)$ of week 2 and (**d**) for $\chi(:,:,3)$ of week 3.

2.4.3. The Tucker and CANDECOMP/PARAFAC Models

When classic matrix factorization approaches are used, part of the multidimensional structure of the data is lost due to the collapse of some of the tensor's modes for framing matrices. However, tensor decomposition techniques allow the explicit exploitation of the multidimensional data structure. Among many existing tensor decomposition techniques, there are two main tensor approaches, which are the ones considered in this work: the Tucker decomposition and the canonical decomposition (CP) with parallel factors, also known as (CANDECOMP/PARAFAC) [11–14,21]. In fact, the CP is a particular case of Tucker decomposition. For simplicity reasons, and following the particular case considered in this work, these decompositions are shown for an order of 3. The Tucker model proposed in [18] decomposes a third-order tensor $\chi^{I\times J\times K}$ as a multilinear transformation of a typically smaller core tensor $G^{L\times M\times N}$ by the factor matrices $\boldsymbol{A}^{I\times L}$, $\boldsymbol{B}^{J\times M}$ and $\boldsymbol{C}^{K\times N}$, accounting for the linear interactions between each of the mode's components. It is common to use the notation Tucker(L,M,N) to indicate the number of vectors that belong to each mode. Using the tensor–matrix product, this decomposition can be written as

$$\chi^{I\times J\times K} \approx G^{L\times M\times N} \times_1 \boldsymbol{A}^{I\times L} \times_2 \boldsymbol{B}^{J\times M} \times_3 \boldsymbol{C}^{K\times N}. \tag{10}$$

The CP decomposition is a particular case of the Tucker decomposition where the number of vectors of each core dimension is the same, that is $L = M = N$ (=D). The interactions in CP are only between columns of the same indices, implying that the core tensor $D^{D\times D\times D}$ is diagonal and, therefore, the only non-zero elements are in the main diagonal (i.e., $d_{l,m,n} \neq 0$, if and only if $l = m = n$). The CP decomposition was independently proposed in [16,19,20]. The CP model in terms of the tensor–matrix product can be written as

$$\mathcal{X}^{I\times J\times K} \approx D^{D\times D\times D} \times_1 \boldsymbol{A}^{I\times D} \times_2 \boldsymbol{B}^{J\times D} \times_3 \boldsymbol{C}^{K\times D}. \tag{11}$$

The same CP model can be given in terms of the outer product of the three vectors $\mathbf{a}_i$, $\mathbf{b}_i$, and $\mathbf{c}_i$ as

$$\mathcal{X}^{I\times J\times K} \approx \sum_{i=1}^{D} \lambda_i \mathbf{a}_i \circ \mathbf{b}_i \circ \mathbf{c}_i, \tag{12}$$

where the column vectors $\mathbf{a}_i$, $\mathbf{b}_i$ and $\mathbf{c}_i$ are related to the matrices of Equation (11) according to $\boldsymbol{A}^{I\times D} = [\mathbf{a}_1 \cdots \mathbf{a}_D]$, $\boldsymbol{B}^{J\times D} = [\mathbf{b}_1 \cdots \mathbf{b}_D]$, and $\boldsymbol{C}^{K\times D} = [\mathbf{c}_1 \cdots \mathbf{c}_D]$. Figure 6a graphically shows the Tucker(L,M,N) decomposition in terms of the tensors involved and the factor matrices for the three-dimensional case, while Figure 6b shows the CP(D) decomposition, first in terms of the tensor–matrix product and then, using the outer product of the vectors.

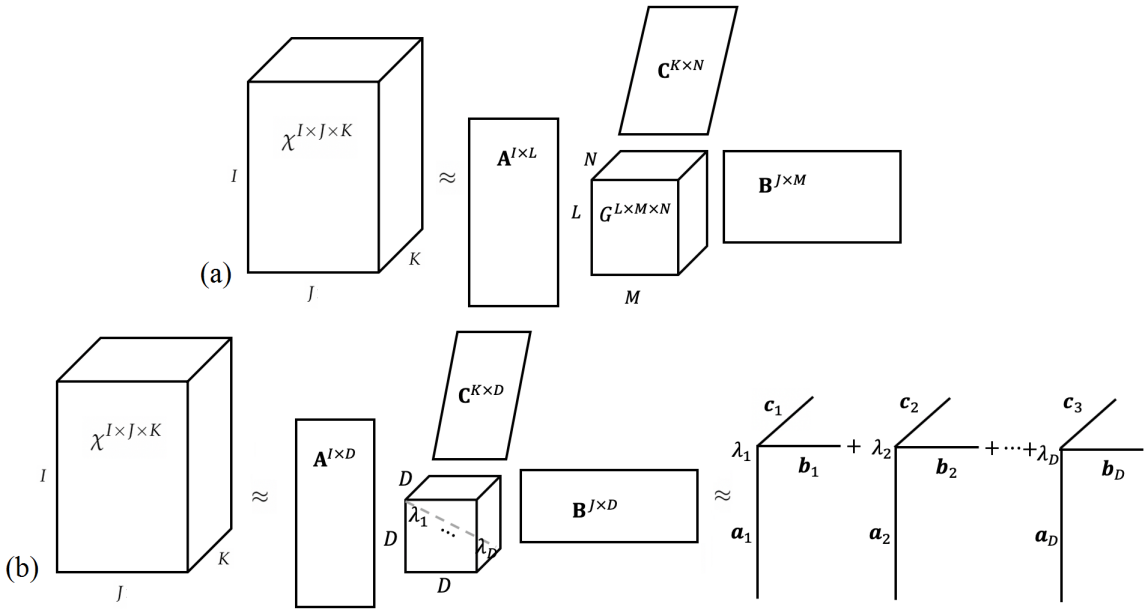

Figure 6. (**a**) Tucker model Tc(L,M,N); (**b**) CANDECOMP/PARAFAC model CP(D) .

2.4.4. CP Weighted Optimization (CP-Wopt)

In [9], a modification of the CP decomposition was used in the presence of missing data by modeling only the known entries and ignoring (marginalizing) the missing ones. The algorithm uses a first-order optimization approach to solve the weighted least-squares problem, and it is known as CP-Wopt (CP Weighted OPTimization). CP-Wopt is one of the most widely used data completion algorithms, and it was proven to be extremely useful in recovering missing data when compared with two-dimensional methods [9]. The CP-Wopt algorithm is included in the Poblano library [27]. Our first option was to use it for our particular data set for these reasons. To test the performance of the method and to be able to quantify it, it was used with known values that had been intentionally eliminated. From the results obtained, we extracted two main insights:

- The algorithm returns a tensor with the missing data filled in but with slightly different values at the continuity points at the ends of the lost burst. When we used the recovered data to fill in the original tensor, this discontinuity usually made the recovery results given by the *ramp* or *FIR* methods worse.
- It was observed that the recovered data followed the variations of the original signal in a very reliable way, although with an offset, since the use of tensors can exploit the inter-relationships between dimensions. This behaviour persisted as the size of the lost bursts increased, surpassing the performance of the FIR predictors.

An example of the way the CP-Wopt algorithm works is shown in Figures 7 and 8 for tensor $\chi^{288\times7\times3}$ with randomly eliminated bursts (shown in red). It can be appreciated that the estimation (in cyan) occasionally differs from the original data (in blue), although it reproduces original fluctuations. Figure 7 shows good results regarding the completion of a small size tensor $\chi^{288\times7\times3}$ with a burst of 100 missing samples using the CP-Wopt algorithm. In Figure 8, it can be observed that sometimes, if the corresponding cyan color section (output of CP-Wopt) is used directly to fill the empty positions, it not only makes a considerable error, but the continuity at the burst extremes is also lost. However, the fluctuation of the signal still remains despite this *offset*.

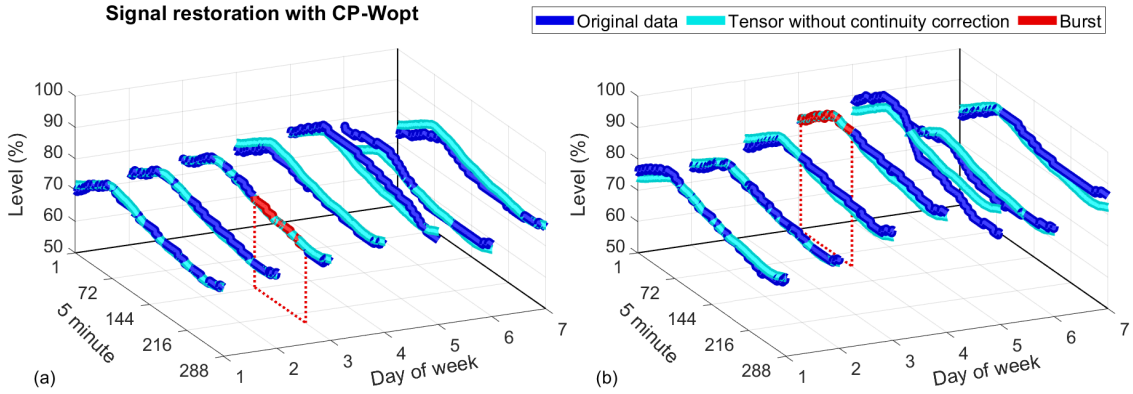

Figure 7. A particular case to show the performance of the CP Weighted OPTimization (CP-Wopt) algorithm when recovering two missing bursts in a $\chi^{288\times7\times3}$ tensor. The original signal is shown in blue, the section corresponding to the erased burst is in red, and the result of the algorithm is displayed in cyan. (**a**) Shows the data in $\chi(:,:,1)$ corresponding to week 1, and (**b**) shows the data in $\chi(:,:,2)$ corresponding to week 2. It can be observed that the output of the CP-Wopt algorithm follows the variations (the first derivative) of the original data, although with some points differing.

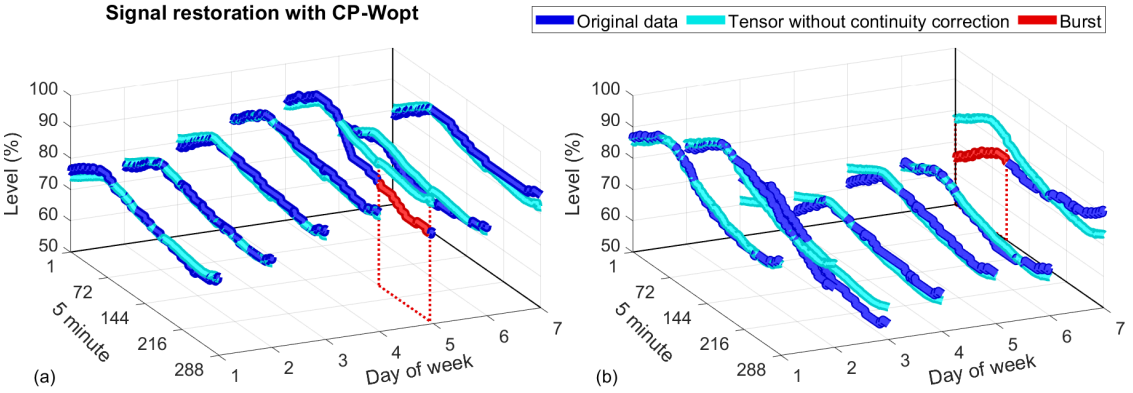

Figure 8. Representation of the offset error that the CP-Wopt algorithm sometimes performs when recovering two missing bursts in a $\chi^{288\times7\times3}$ tensor. The original signal is shown in blue, the section corresponding to the erased burst of 100 samples is shown in red, and the result of the algorithm is shown in cyan: (**a**) shows the data in $\chi(:,:,2)$ corresponding to week 2; (**b**) shows the data in $\chi(:,:,3)$ corresponding to week 3.

2.4.5. The Proposed Tensor Method with an Offset Correction

The algorithm proposed to fill the lost data takes advantage of the inter-relationships between dimensions that capture the algorithms based on tensors. Thus, the signal is tensorized, as explained in Section 2.4.2, in such a way as to take advantage of the patterns between the days of the week and between the weeks. To do this, the following steps are established:

- One of the linear methods for data completion described in Section 2.3 is used to fill the empty burst. The positions of the lost data are stored. After this step, there are no empty elements in the tensor.
- A low-range tensor approximation is performed using tensor decomposition. In our case, the popular Tucker and CP algorithms were explored, as explained in Section 2.4.3. This simple approach captures the most important variations of the original tensor, reproducing it with a high level of accuracy. The difference between the low-rank approximation and the original tensor consists of high-frequency components with low amplitudes.
- Samples $\hat{x}_i$ found in the original lost data positions are retrieved from the low-range tensor and corrected as explained below to maintain continuity when the burst ends. The correction procedure carried out is similar to the one applied to the *FIR* method. The resulting compensated data are the ones that are subsequently assigned. This last step is referred to as an offset correction.

To explain the data correction process used with the aim of preserving the continuity between the extremes, a burst of lost data of N samples is first considered. We let x_a be the last reliable received sample before the loss occurs and x_b be the first sample correctly received once the burst is over. Let $\hat{x}_i$, $i = 1, \cdots, N$ be the estimates made, regardless of the method used, corresponding to the positions where the data were missed. Let $\hat{x}_0$ and $\hat{x}_{N+1}$ be the estimates of the positions corresponding to x_a and x_b. The method that produces the offset is $O_a = x_a - \hat{x}_0$, at the beginning, and $O_b = x_b - \hat{x}_{N+1}$, at the end. Thus, to maintain continuity, the correction of the offset is performed according to the expression

$$\text{for} \quad i = 1 \ldots N; \qquad \tilde{x}_i = \begin{cases} \hat{x}_i - \dfrac{O_a + O_b}{2} & \text{if } N = 1 \\ \hat{x}_i - \dfrac{(N-i)O_a + (i-1)O_b}{N-1} & \text{if } N > 1 \end{cases} \tag{13}$$

where $\tilde{x}_i$ are the estimations with the corrected offset.

2.5. Algorithm Performance Evaluation

To be able to evaluate the algorithms and compare them, a loss-less stretch of data was selected. This data set was used as a reference, and bursts of L values were intentionally removed randomly. This way, when an algorithm fills the burst, it was possible to compare the results with the original values and thus establish an objective measurement. For each burst, the mean squared error (MSE) per sample was chosen according to the following expression

$$MSE = \frac{1}{L} \sum_{i=1}^{L} \sqrt{(x_i - \hat{x}_i)^2}. \tag{14}$$

One thousand different positions corresponding to the starts of the burst were randomly calculated to test the algorithms under the same conditions. These positions and L were saved and then used to generate the bursts in all tests. Therefore, each algorithm reconstructed the same 1000 bursts to compute the final score (the MSE per sample).

3. Results

3.1. Tuning the FIR Filters

Among the three methods used that do not involve tensors, the only one that is configurable is the method based on the FIR filters. For the first experiment, a test was performed to determine the right combinations of L and M. The experiment involved randomly erasing data bursts and recovering them with the *FIR* method while a sweep of both parameters was performed. The results were evaluated by calculating the mean squared error (MSE) made per sample. The bursts removed had a length of 100 samples. The results, which can be seen in Figure 9, are the average of 1000 experiments. The combination of parameters selected as the best option was ($L = 65$, $M = 2000$), which achieved the use of minimum values for M and L, and the MSE was, at most, 1% bigger than the minimum error. Three more combinations from both parameters were tested using similar criteria. The rule of choosing the minimum M and L values was followed. In addition, a restriction was imposed that stated that the MSE was, at most, 5%, 10%, or 15% bigger than the minimum error, respectively.

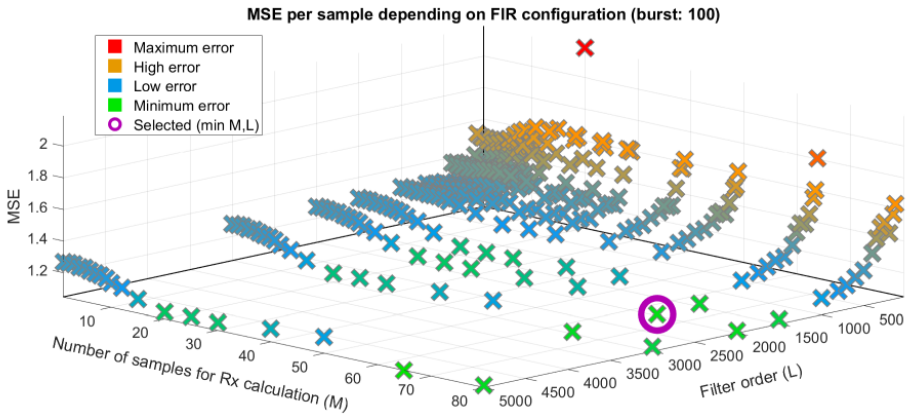

Figure 9. Mean square error (MSE) per sample of the *FIR* method depending on the FIR order (L) and the number of samples used in the auto-correlation calculation (M).

3.2. Tensor Methods—Best Factorization Size

This aim of this experiment was to determine the effects of the decomposition size on the proposed tensor-based methods. Thus, a set of data was *tensorized* in a tensor $\chi^{288\times7\times3}$ of size $288 \times 7 \times 3$, and the performance of the different algorithms were evaluated by randomly generating a burst of 1000 lost data points according to the system described in Section 2.5. Since two popular decompositions were explored, the experiments are repeated for each type of decomposition.

3.2.1. Algorithms employing the Tucker factorization

The algorithms used to perform the initial imputation of values were evaluated by means of the *last value*, the *ramp*, and certain different configurations of the *FIR* methods, followed by tensorization and corresponding Tucker decomposition of different sizes. For each case, the effect of applying or not applying the offset correction was considered. For reference purposes, the results were compared with (1) the value imputation method used directly without performing the tensorization step and (2) with the original CP-Wopt method and its modified version with the offset correction. The results are summarized using bar graphs in Figure 10. In this figure, the bar diagrams are segregated into subfigures according to the method of imputation used. In the axis of abscissas, the decomposition used is shown in parenthesis. When the imputation method used was the *FIR*, the method is referenced in the graphs as FIR (L,M), with the first number being the length of the filter and the second being the number of values used to calculate the auto-correlation coefficients. For each case, the bar of the decomposition with the minimum MSE per sample is marked in red (as well as certain other bars with

similarly low values of MSE). In addition, a red horizontal line marks the specific value that is taken as a reference for comparison with the others. In order to achieve statistically representative results, the MSE values are the result of averaging 1000 experiments. Note that all tested methods performed their best with the Tucker(3,3,1) decomposition.

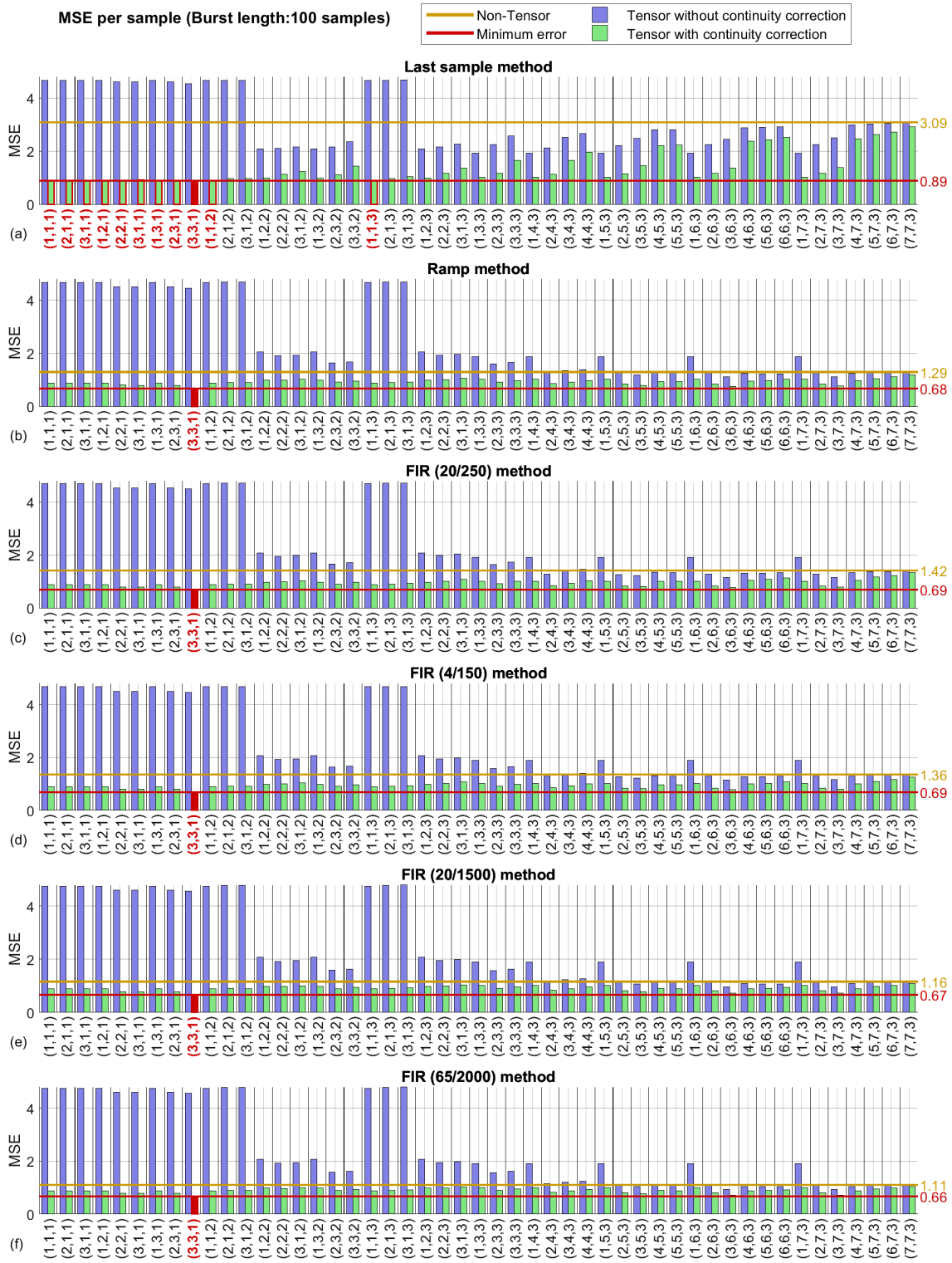

Figure 10. MSE per sample of the Tucker decomposition depending on the core tensor. The MSE of the restoration is shown with the continuity correction and without it. The bars marked in red have an error no bigger than 5% of the minimum.

3.2.2. Algorithms Employing CANDECOMP/PARAFAC or CP Factorization

A similar test was performed for the CP decomposition. In this case, only D was explored. The results are also summarized using bar graphs in Figure 11. The order of the decompositions ranged from 2 to 15. In Figure 11, for each experiment, the decomposition that provided the lowest MSE per sample again has the corresponding bar marked in red. Additionally, a horizontal line is drawn with the level of this bar to allow for comparison with the performance of the data imputation method (i.e., non-tensor method) before decomposition, which is marked with an orange horizontal line. The difference between these two lines illustrates the improvement of the method in terms of the MSE per sample. It can also be observed that the size of the decomposition that provides the best results is around $D = 6$, except for the *last value* method, which gives the best results when $D = 2$ and gets worse as the size of the decomposition increases.

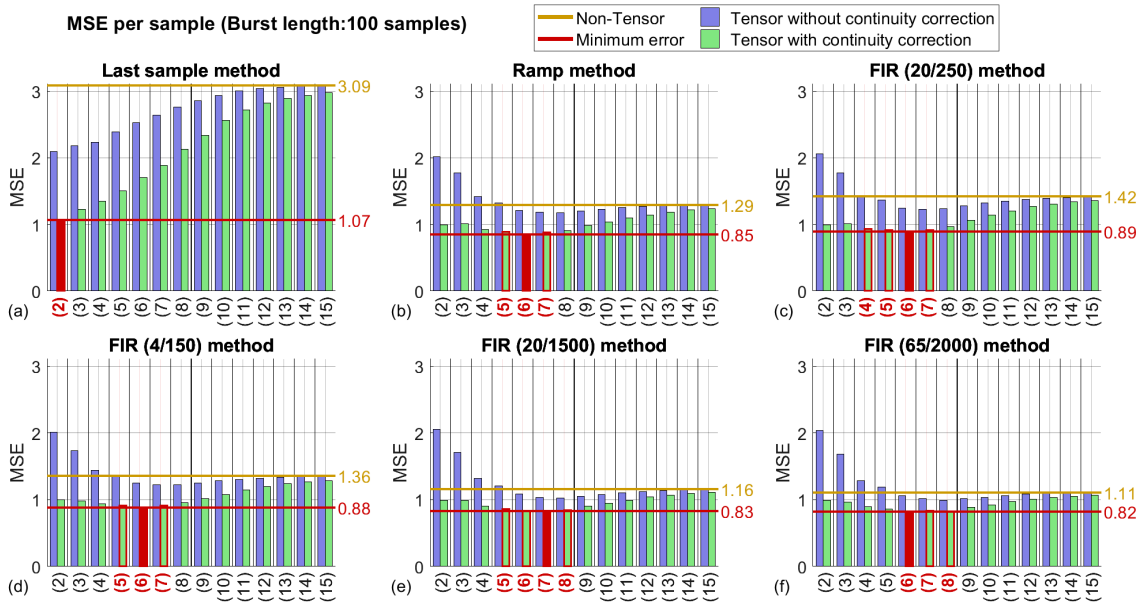

Figure 11. MSE per sample of the CP decomposition depending on the core tensor. The MSE of the restoration is shown with the continuity correction and without it. The bars marked in red have an error of no bigger than 5% of the minimum.

Figure 12 displays the results of different imputation methods using a selection of CP and Tucker decompositions that function more effectively, in particular, Tucker(2,2,1), Tucker(3,3,1), CP(2), and CP(6). In addition to these, the Tucker(1,1,1) decomposition, which is equivalent to the CP(1) (as can be intuitively observed in Figure 6) has also been included. Finally, the CP-Wopt is also shown, which works directly with the missing values and does not depend on the first stage of the restoration method.

Thus, we selected Tucker(1,1,1) = CP(1) (which are the same), CP(2), CP(6), Tucker(2,2,1), and Tucker(3,3,1) as references to evaluate the different algorithms. Figure 12 was constructed using the same experimental conditions, tenso,r and lost burst sizes as those used in the experiments summarized in the two previous figures.

Note that each algorithm is presented with and without an offset correction. As explained previously, for each method, the sample that produces the lowest MSE is marked in red, and a horizontal line is plotted at this level to facilitate comparisons. It can be seen that once again, for all the methods shown, the decomposition Tucker(3,3,1) obtains the best results (see Figures 10 and 12), except for the *last sample* value imputation method. Nevertheless, the remaining decompositions also achieve very similar results in a systematic way.

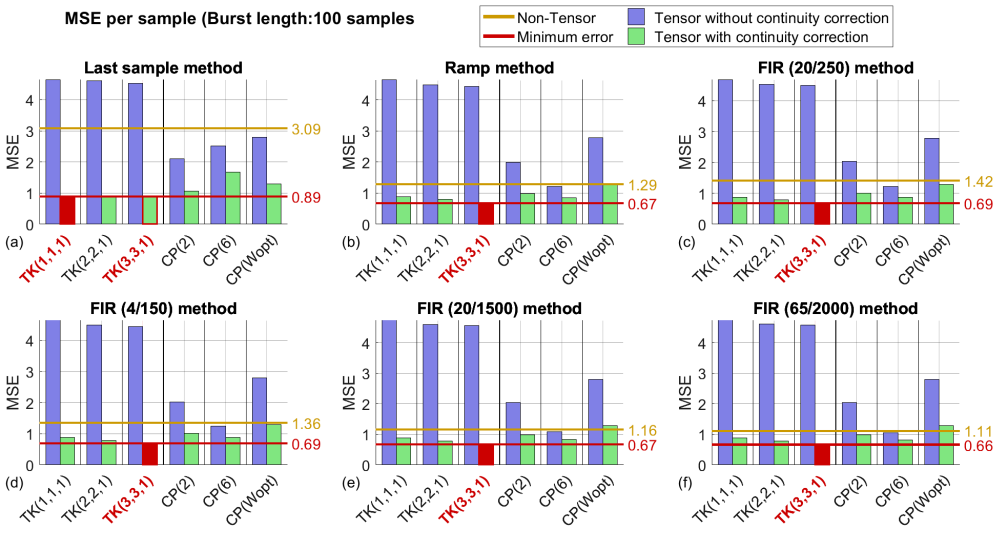

Figure 12. MSE of different value imputation methods. The results are the mean of 1000 simulations. The number of samples in the burst is 100. The position of the the burst is randomly selected. The tensor contains three weeks worth of data (288,7,3).

3.3. The Algorithm's Performance According to the Tensor Size

Once the structure of the tensor has been defined by arranging the data in five-minute, daily and weekly intervals, it is possible to change the tensor size by taking into account either more or fewer weeks. Hence, the effect of the tensor size on the recovery of the lost burst should be investigated. The experiment carried out was as follows. A burst of $N = 100$ was randomly erased from a known tensor that packs data from a different number of weeks. The same starting positions for the bursts obtained in previous experiments for the $\chi^{288\times7\times3}$ tensor were used and the performance test was run. Then, a week of known data was progressively added at the beginning and end of the tensor, and the experiment was reproduced. The configuration $\chi^{288\times7\times2i+1}$ where $i = 1$ was used as the original configuration, and $\chi^{288\times7\times31}$ where $i = 15$ was evaluated.

The results for the selected algorithm configurations are shown in Figure 13.

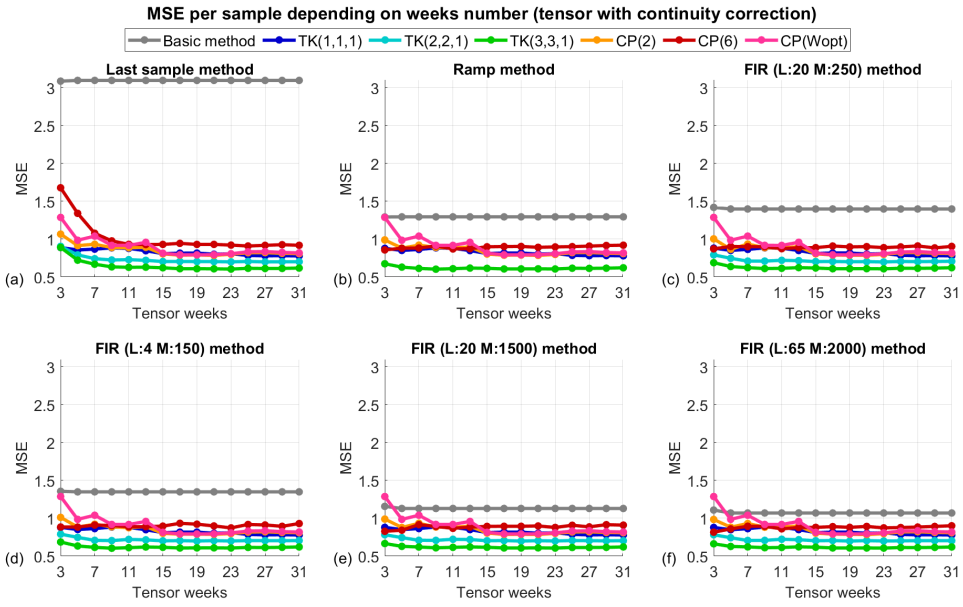

Figure 13. Trend of the MSE with an increaasing number of weeks configured in the tensor. The results are the mean of 1000 simulations. The tensor contains three weeks worth of data (288,7,3). The number of samples in the burst is 100, and the position of the burst was randomly selected.

3.4. The Algorithm's Performance According to the Length of the Missing Data Burst

Figure 14 shows the evolution of the selected algorithms according to the size of the burst, as well as the behavior of the restoration algorithm with the continuity correction using different configurations of factorization and for different "non-tensor" initial methods. Thus, the different imputation methods that alter the burst size (L = 25, 50, 100, 150, 200 and 250) were evaluated. Values of $L > 250$ imply a duration of nearly a day or more, and in this case, are challenging to find because the maintenance department has had time to repair the sensor. For the experiment, the same method of selecting the starting points of the burst as that described in the previous experiments was followed and the data were erased from an $\chi^{288\times7\times3}$-sized tensor. The behavior of the CP-Wopt method with subsequent offset correction is also shown. To facilitate a comparison, the axes of all the sub-graphics are equal.

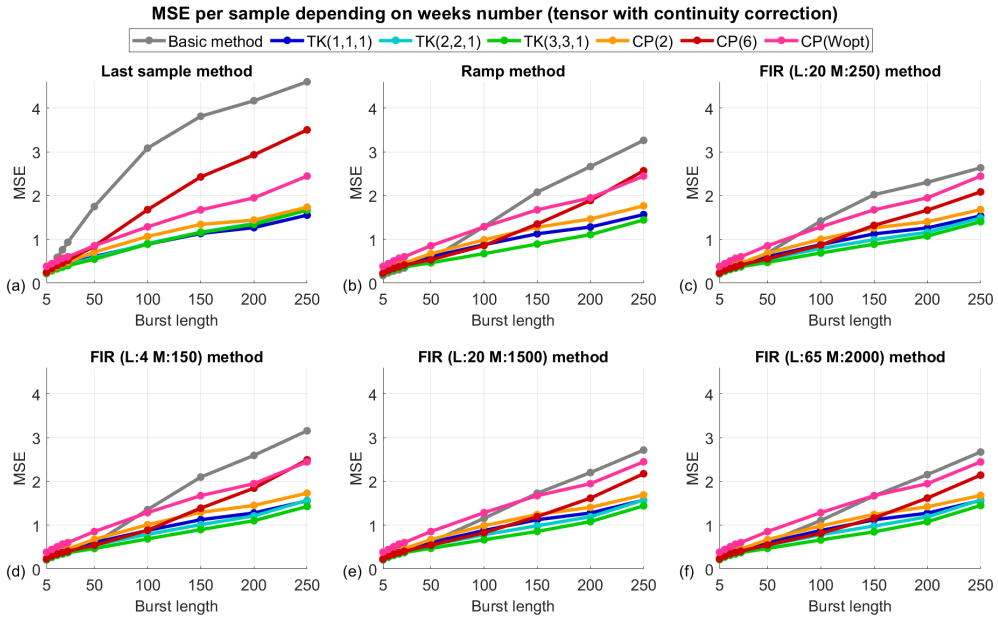

Figure 14. Trend of the MSE with a progressively increasing number of samples. The results are the mean of 1000 simulations. The tensor contains three weeks worth of data (288,7,3), and the position of the first missing sample of the burst is randomly selected.

4. Discussion and Conclusions

Nowadays, SCADA systems store a huge amount of data from a wide variety of industrial processes. The vast amount of stored data can be used to improve plant models for prognosis and diagnosis purposes. However, the first step is to ensure that the available data are reliable, as it is a major challenge to guarantee the consistency of the rest of the steps and procedures. This paper put forward a specifically derived method to recover lost data that occur in bursts, due to failure of either a sensor or the link used to transmit the measurements. In the problem addressed, the signal of interest has soft variations due to the inertia of the process behind it and the simplicity of the sensors which causes the signal to have a staggered appearance, as shown in Figure 5. The data are recorded in five minute intervals, and although the methods based on interpolation and prediction work quite well, tensorization of the signal takes advantage of the specific patterns on daily and weekly scales. This advantage becomes more significant as the size of the missing burst increases, as shown in Figure 14. Among the different strategies for recovering lost data is the allocation of values using more or less elaborate methods, such as those based on interpolation and prediction. Value imputation techniques involve the assignment of values to missing data without using tensor formulation. We compared three such techniques, two of which were very simple but that have been used historically by the water company to solve this problem, and a third, more elaborate one, based on the combination

of the forward and backward linear estimations. FIR filters are used as predictors. Their optimal lengths and the method they employ to estimate the auto-correlation coefficients to calculate the best configuration according to the Wiener formulation were studied and detailed. The use of tensor algebra makes it possible to exploit the relationships between data dimensions. Among tensor strategies, marginalization techniques avoid assumptions being made about the missing data. For certain types of signals and for certain types of missing data distributions, marginalization techniques work remarkably well [9]. For instance, the CP-Wopt is able to recover the correct underlying components from noisy data with up to 99% of missing data for third-order tensors; in contrast, two-way methods become unstable with missing data levels of only 25–40% [9]. However, for some applications, the recovery of missing data can be improved by using a specialized solution, such as in the presented case or the one in [43]. In the present work, in which the missing data were lost in bursts, combining an imputation method followed by a low-range tensor approximation with an ad-hoc offset correction to ensure continuity in the extremes of the missing burst produced a better performance over all the other tested completion methods. We developed a procedure to compare the performance of the algorithms. For the proposed approach, different configurations for the predictor were explored (Figure 9), the optimal size of tensor decomposition was determined (Figures 10 and 11), and the dimensions of the tensor that offer the best performance were obtained (Figure 13). We also compared algorithms by varying the lengths of the missing bursts (Figure 14).

Various conclusions can be drawn from the results. In the general case of very short bursts, when $N \leq 20$, all methods make similar errors. Using a rough, less precise imputation method in the first stage usually works best with very low-range decompositions. This means that for the *last sample* method, the best performance is obtained using the most elementary tensor decompositions TK(1,1,1) and CP(1). However, the *last sample* method degenerates very quickly if the burst is larger than 25 samples. If the imputation method is more sophisticated, a bigger-rank decomposition improves the performance of the algorithm. We found the CP(6) (CANDECOM/PARAFAC) and Tucker(3,3,1) to be good options in all cases. The two decompositions work very similarly, although the latter may be slightly superior. Concerning the optimal size of the tensor, we explored the third dimension, which corresponds to weekly measurement, so that for the same removed burst, the tensor with known data was increased. We first ran the algorithms with three weeks of data and then proceeded to progressively add weeks to evaluate if the recovery improved with additional known data. The results showed an improvement in the performance until approximately seven weeks, when the performance results stabilized. When comparing the different algorithms with lost bursts of different lengths, it is apparent that for burst sizes of up to $N = 100$ samples, the results obtained with the *ramp* method are as good as those obtained using predictors, and it would only be justified to use a more complex method when N is (approximately) greater than 100. Furthermore , in all the experiments where algorithms were compared, we introduced the original CP-Wopt completion method and a modification of it in which the offset is corrected in the same way as that performed for the rest of the proposed algorithms. Finally, we note that in all cases, the simple offset correction step plays a fundamental role in improving the performance. We illustrated that the offset correction works by looking at examples of a couple of particular cases. The different steps of the proposed method are represented in Figures 15 and 16. i.e., to allow a comparison to be made, the results of the algorithm of value imputation, the reconstruction of the signal through a low-range tensor, and the value of the reconstruction after correcting the offset, are visualized together with the data of the original burst that were erased at the beginning of the experiment.

An additional advantage of the method is that the execution time is not excessive. The execution time of the different parts of the algorithm was measured performed on an Intel(R) Core(TM) i5-6200U, 2.3 GHz platform with 8 GB of RAM with Windows 7 Professional and Matlab 2018b. The basic methods of imputation (last sample and ramp) only took 0, 2, and 4 ms. The FIR method required greater computation time, from 2.5 to 10 s, mainly due to auto-correlations and pseudo-inverses. Tensor decomposition took between 0.1 and 0.5 s.

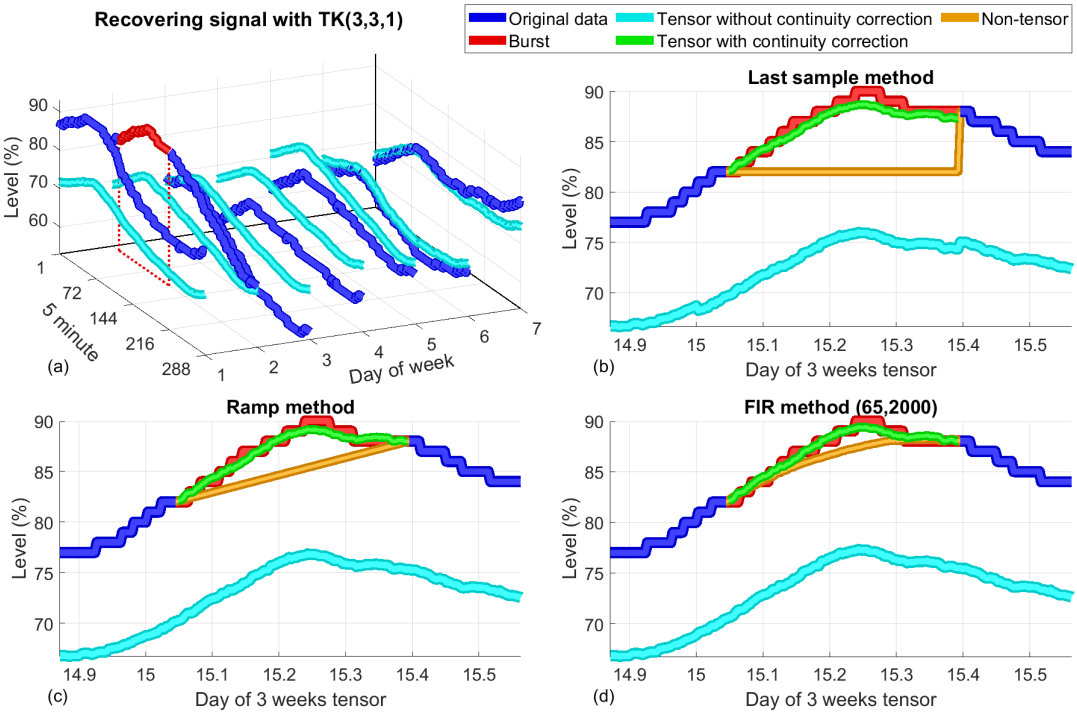

Figure 15. An example of the restoration stages. The tensor contains three weeks worth of data (288,7,3). The burst is 100 samples long, and its position is randomly selected: (**a**) shows the week of the tensor where the burst is located, $\chi(:,:,3)$, corresponding to week 3. Parts (**b**–**d**) show the three steps of the restoration process for the *last sample*, the *ramp*, and the *FIR* methods, respectively.

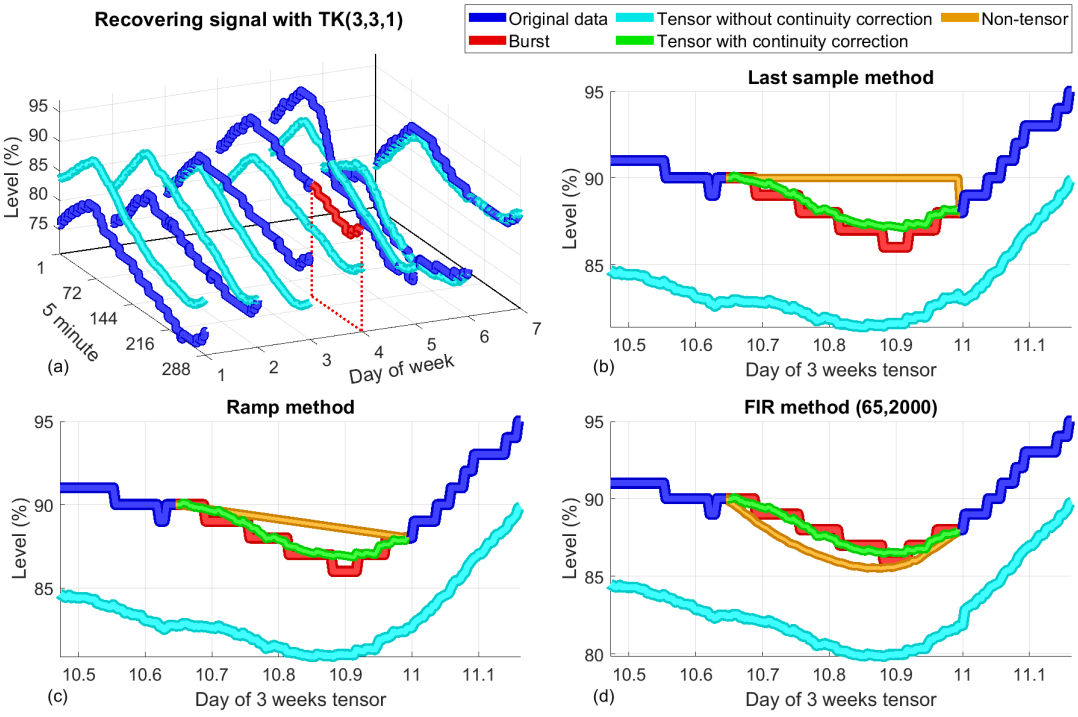

Figure 16. An example of the restoration stages. The tensor contains three weeks worth of data (288,7,3). The burst is 100 samples long, and its position is randomly selected: (**a**) shows the week of the tensor where the burst is located, $\chi(:,:,1)$, corresponding to week 2. Parts (**b**–**d**) show the three steps of the restoration process for the *last sample*, the *ramp*, and the *FIR* methods, respectively.

Finally, it is worth noting that SCADA systems operating in any field in which data losses occur in bursts could also use the proposed method to complete stored data.

Author Contributions: Conceptualization, P.M.-P., M.S.-S and A.M.-S.; methodology, P.M.-P. and M.S.-S; software, A.M.-S. and P.M.-P. ; validation, P.M.-P., M.S.-S and A.M.-S.; formal analysis, P.M.-P. and M.S.-S; investigation, P.M.-P. and A.M.-S.; resources, M.S.-S; data curation, A.M.-S.; writing–original draft preparation, P.M.-P. and A.M.-S.; writing–review and editing, P.M.-P., M.S.-S and A.M.-S.; supervision, P.M.-P. and M.S.-S; funding acquisition, P.M.-P. and M.S.-S.

Funding: Financial support by the Agency for Management of University and Research Grants (AGAUR) of the Catalan Government to Arnau Martí-Sarri is gratefully acknowledged.

Acknowledgments: We thank the company Aigües de Vic S.A. for giving us access to their databases to perform this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Langhammer, J.; Česák, J. Applicability of a Nu-Support Vector Regression Model for the Completion of Missing Data in Hydrological Time Series. *Water* **2016**, *8*, 560. [CrossRef]
2. Ahlheim, M.; Frör, O.; Luo, J.; Pelz, S.; Jiang, T. Towards a Comprehensive Valuation of Water Management Projects When Data Availability Is Incomplete—The Use of Benefit Transfer Techniques. *Water* **2015**, *7*, 2472–2493. [CrossRef]
3. Zhao, Q.; Zhu, Y.; Wan, D.; Yu, Y.; Cheng, X. Research on the Data-Driven Quality Control Method of Hydrological Time Series Data. *Water* **2018**, *10*, 1712. [CrossRef]
4. Ekeu-wei, I.T.; Blackburn, G.A.; Pedruco, P. Infilling Missing Data in Hydrology: Solutions Using Satellite Radar Altimetry and Multiple Imputation for Data-Sparse Regions. *Water* **2018**, *10*, 1483. [CrossRef]
5. Lamrini, B.; Lakhal, E.K.; Le Lann, M.V.; Wehenkel, L. Data validation and missing data reconstruction using self-organizing map for water treatment. *Neural Comput. Appl.* **2011**, *20*, 575–588. [CrossRef]
6. Blanch, J.; Puig, V.; Saludes, J.; Quevedo, J. Arima models for data consistency of flowmeters in water distribution networks. *IFAC Proc. Vol.* **2009**, *42*, 480–485. [CrossRef]
7. Puig, V.; Ocampo-Martinez, C.; Pérez, R.; Cembrano, G.; Quevedo, J.; Escobet, T. *Real-Time Monitoring and Operational Control of Drinking-Water Systems*; Springer: Basel, Switzerland, 2017.
8. Cugueró-Escofet, M.À.; García, D.; Quevedo, J.; Puig, V.; Espin, S.; Roquet, J. A methodology and a software tool for sensor data validation/reconstruction: Application to the Catalonia regional water network. *Control Eng. Pract.* **2016**, *49*, 159–172. [CrossRef]
9. Acar, E.; Dunlavy, D.M.; Kolda, T.G.; Mørup, M. Scalable tensor factorizations for incomplete data. *Chemom. Intell. Lab. Syst.* **2011**, *106*, 41–56. [CrossRef]
10. Signoretto, M.; Van de Plas, R.; De Moor, B.; Suykens, J.A. Tensor versus matrix completion: A comparison with application to spectral data. *IEEE Signal Process. Lett.* **2011**, *18*, 403. [CrossRef]
11. Mørup, M. Applications of tensor (multiway array) factorizations and decompositions in data mining. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2011**, *1*, 24–40. [CrossRef]
12. Kolda, T.G.; Bader, B.W. Tensor decompositions and applications. *SIAM Rev.* **2009**, *51*, 455–500. [CrossRef]
13. Cichocki, A.; Mandic, D.; De Lathauwer, L.; Zhou, G.; Zhao, Q.; Caiafa, C.; Phan, H.A. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE Signal Process. Mag.* **2015**, *32*, 145–163. [CrossRef]
14. Comon, P. Tensors: A brief introduction. *IEEE Signal Process. Mag.* **2014**, *31*, 44–53. [CrossRef]
15. Vaseghi, S.V. *Advanced Digital Signal Processing and Noise Reduction*; John Wiley & Sons: Chichester, UK, 2008.
16. Harshman, R.A. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis. In *Working Papers in Phonetics*; UCLA: Los Angeles, CA, USA, 1970; Volume 16, pp. 1–84.
17. Sørensen, M.; Lathauwer, L.D.; Comon, P.; Icart, S.; Deneire, L. Canonical polyadic decomposition with a columnwise orthonormal factor matrix. *SIAM J. Matrix Anal. Appl.* **2012**, *33*, 1190–1213. [CrossRef]
18. Tucker, L.R. Some mathematical notes on three-mode factor analysis. *Psychometrika* **1966**, *31*, 279–311. [CrossRef]
19. Carroll, J.D.; Chang, J.J. Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. *Psychometrika* **1970**, *35*, 283–319. [CrossRef]
20. De Lathauwer, L.; De Moor, B.; Vandewalle, J. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.* **2000**, *21*, 1253–1278. [CrossRef]

21. Sidiropoulos, N.D.; De Lathauwer, L.; Fu, X.; Huang, K.; Papalexakis, E.E.; Faloutsos, C. Tensor decomposition for signal processing and machine learning. *IEEE Trans. Signal Process.* **2017**, *65*, 3551–3582. [CrossRef]
22. Liu, J.; Musialski, P.; Wonka, P.; Ye, J. Tensor completion for estimating missing values in visual data. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 208–220. [CrossRef] [PubMed]
23. Roughan, M.; Zhang, Y.; Willinger, W.; Qiu, L. Spatio-temporal compressive sensing and internet traffic matrices. *IEEE/ACM Trans. Netw.* **2012**, *20*, 662–676. [CrossRef]
24. Wang, L.; Xie, K.; Semong, T.; Zhou, H. Missing Data Recovery Based on Tensor-CUR Decomposition. *IEEE Access* **2018**, *6*, 532–544. [CrossRef]
25. Gandy, S.; Recht, B.; Yamada, I. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Probl.* **2011**, *27*, 025010. [CrossRef]
26. Zhao, Q.; Zhang, L.; Cichocki, A. Bayesian CP factorization of incomplete tensors with automatic rank determination. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1751–1763. [CrossRef] [PubMed]
27. Dunlavy, D.M.; Kolda, T.G.; Acar, E. *Poblano v1.0: A Matlab Toolbox for Gradient-Based Optimization*; Technical Report SAND2010-1422; Sandia National Laboratories: Albuquerque, NM, USA; Livermore, CA, USA, 2010.
28. Bader, B.W.; Kolda, T.G.; others. MATLAB Tensor Toolbox Version 2.6. February 2015. Available online: http://www.sandia.gov/~tgkolda/TensorToolbox/index-2.6.html (accessed on 11 November 2018).
29. Vaidyanathan, P. The theory of linear prediction. In *Synthesis Lectures on Signal Processing*; Morgan & Claypool Publishers: San Rafael, CA, USA, 2007; Volume 2, pp. 1–184.
30. Kailath, T.; Sayed, A.H.; Hassibi, B. Linear Estimation. In *Number Book*; Prentice Hall: Upper Saddle River, NJ, USA, 2000.
31. Mitter, S. Linear Estimation-T. Kailath, AH Sayed, and B. Hassibi. *IEEE Trans. Autom. Control* **2003**, *48*, 177–182.
32. Wang, Z.; Yang, F.; Ho, D.W.; Liu, X. Robust finite-horizon filtering for stochastic systems with missing measurements. *IEEE Signal Process. Lett.* **2005**, *12*, 437–440. [CrossRef]
33. Humpherys, J.; Redd, P.; West, J. A fresh look at the Kalman filter. *SIAM Rev.* **2012**, *54*, 801–823. [CrossRef]
34. Junninen, H.; Niska, H.; Tuppurainen, K.; Ruuskanen, J.; Kolehmainen, M. Methods for imputation of missing values in air quality data sets. *Atmos. Environ.* **2004**, *38*, 2895–2907. [CrossRef]
35. Quinteros, M.E.; Lu, S.; Blazquez, C.; Cárdenas-R, J.P.; Ossa, X.; Delgado-Saborit, J.M.; Harrison, R.M.; Ruiz-Rudolph, P. Use of data imputation tools to reconstruct incomplete air quality datasets: A case-study in Temuco, Chile. *Atmos. Environ.* **2019**, *200*, 40–49. [CrossRef]
36. Yang, Y.; Ma, J.; Osher, S. Seismic data reconstruction via matrix completion. *Inverse Probl. Imaging* **2013**, *7*, 1379–1392. [CrossRef]
37. Box, G.E.; Jenkins, G.M. *Time Series Analysis: Forecasting and Control*; Revised ed.; Holden-Day: San Francisco, CA, USA, 1976.
38. Zhang, Z.; Ely, G.; Aeron, S.; Hao, N.; Kilmer, M. Novel methods for multilinear data completion and de-noising based on tensor-SVD. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 3842–3849.
39. Zhang, Z.; Aeron, S. Exact Tensor Completion Using t-SVD. *IEEE Trans. Signal Process.* **2017**, *65*, 1511–1526. [CrossRef]
40. Filipović, M.; Jukić, A. Tucker factorizatio with missig data with applicatio to low-rak tesor completio. *Multidimens. Syst. Signal Process.* **2015**, *26*, 677–692. [CrossRef]
41. Yokota, T.; Zhao, Q.; Cichocki, A. Smooth PARAFAC decomposition for tensor completion. *IEEE Trans. Signal Process.* **2016**, *64*, 5423–5436. [CrossRef]
42. Kressner, D.; Steinlechner, M.; Vandereycken, B. Low-rank tensor completion by Riemannian optimization. *BIT Numer. Math.* **2014**, *54*, 447–468. [CrossRef]
43. Sole-Casals, J.; Caiafa, C.F.; Zhao, Q.; Cichocki, A. Brain-Computer Interface with Corrupted EEG Data: A Tensor Completion Approach. *Cognit. Comput.* **2018**, *10*, 1062. [CrossRef]

Apèndix C

EFFECT OF THE DATA TENSORIZATION ON THE RECOVERY OF BURSTS OF MISSING VALUES. AN APPLICATION IN WATER NETWORKS

May 2019

Effect of the data *tensorization* on the recovery of bursts of missing values. An application in water networks

Arnau MARTÍ-SARRI [a,1], Moisès SERRA-SERRA [b] and Pere MARTI-PUIG [a]

[a] *Data and Signal Processing Group, U Science Tech, University of Vic–Central University of Catalonia, c/de la Laura 13, 08500 Vic, Catalonia, Spain*

[b] *MECAMAT Group, U Science Tech, University of Vic–Central University of Catalonia, c/de la Laura 13, 08500 Vic, Catalonia, Spain*

Abstract. The SCADA systems capture a huge quantity of data from different devices. In order to analyze the historical data collected by a specific sensor, in some cases, it is necessary to restore the lost or discarded data. When we deal with a large amount of unknown consecutive samples, this task is more complicated. The study is focused on reconstructing bursts of lost samples of a water reservoir level meter. We prove that the tensors can be a useful mathematical tool to do this function. We know that it is possible to improve the data reconstructions realized with linear methods by applying tensorization techniques. To do this, it is necessary to organize the data in the tensor, searching to take the maximum advantage of the signal periodicity on different levels. In this work, it is verified that reordering the tensor according to the position of the lost samples, that must be recovered, affects the result of the reconstruction. So that, we purpose an optimal tensor ordering for the bursts restoration.

Keywords. water networks, SCADA data, tensor completion, tensor decomposition

1. Introduction

Data analysis is a key element in casting a representative model of a specific system, and to better understand its performance. This also applies to water supply companies, whose water reservoirs need to be modelled in order to find patterns, operating cycles, and, more in general, to tests its performance.

This work focuses on the performance optimization of the main water tank of the Aiguës de Vic SA (AVSA), the water supplier enterprise which operates in the city of Vic, Catalonia. AVSA main deposit shows a capacity of 16.000 m^2 and provides water to both population and companies of Vic. As the demand is continuous, water supply from this reservoir never stops, and purified water coming from the Water Treatment Station (WTS) must be pumped into the deposit according to both demand and tank water level. AVSA adopted a Supervisory Control and Data Acquisition system (SCADA) to

[1] Corresponding Author: Arnau Mart Sarri; E-mail: arnau.marti@uvic.cat

collect data from many different sensors and monitor the water distribution network. This SCADA allows to collect data from the main deposit and store them in a database every 5 minutes. However, it recently appeared that several consecutive groups of data collected from the level meter during the last three years have been lost. The reason may lay in communication breakdowns between the SCADA server and the related Programmable Logic Controller (PLC), or between the latter and the sensor. Given the slow water level variation and the data acquisition frequency over the time, it can be easy to reconstruct a single lost data. Indeed, the sensor measures the percentage of the water level, from 0 to 100%.

However, when dealing with losses of large groups of consecutive data, the reconstruction task can be a challenge. The data validation and completion is usually performed by using traditional approaches that are mainly based on predictive techniques [1] that act reasonably well when the missing data is uniformly distributed or appear in short bursts [1,2,3,4]. However, the performance decreases for longer bursts because they cannot take fully advantage of the hidden structure of the data, when present [5,6]. Notwithstanding, if the data sets show more than two ways of variation, tensor factorization techniques allow to discover the interrelations between dimensions. Among others, the CAN- DECOMP/PARAFAC (CP) [7,8] and the Tucker [9,10,11] tensor decompositions are the most widely known. Nowadays, tensor factorization strategies are extensively applied [12,13,14,15] and the data completion problem takes advantage of them through two main strategies. The first is the *maximization of expectations* which is devoted to assigning estimated values to fill missing data. The second is the marginalization, which avoids missing values in the optimization process [12]. On the previous study, in [16], we demonstrated that the tensors can be a useful tool to manage this type of damaged data by proposing a two-stages methodology. Firstly, the data are restored by linear methods. Secondly, *tensorization* is used to refine the first stage of data reconstruction, improving considerably the results. The tensors allow to take advantage of the signal periodicity to improve the signal reconstruction done by a linear method. Through the careful observation of recovered cases we observed that the way the data are organized in the tensor is crucial to reach successful results.

This article focuses on the way of *tensorize* the data in order to construct the tensor. Furthermore, it demonstrates how is possible to obtain additional improvements in the results by re-configuring the tensor depending on the position of the lost burst. The basic tensor algebra can be found in [17].

2. *Tensorization* procedure

Tensorization is the process of packaging lower-dimensional original data in a container, the tensor, with more dimensions than the original one. The 1-dimensional to 3-dimensional process is shown in Fig. 1 with an example of the both used tensors, the Non-customized and the Customized. The Non-customized tensor is organized considering the week including the burst, as well as indefinite number of weeks before and after. Weeks are programmed to start always on Monday at 00:00. This gives an odd number for the tensor size (3 and 7 weeks, as defined in this research) and the burst is located in the central week of the tensor and can start in any day of this week. As for the Customized tensor, by changing the starting day and hour of the tensor, it forces the burst to

be located in the central week of the tensor, and in the middle of the week. This causes that the weeks will start depending on the burst. The Fig. 1 is the representation of a water deposit level signal during 5 weeks. This sensor gives a percentage of the tank capacity every 5 minutes. When we add new dimensions, we must order the data searching a structure which takes advantage of the signal periodicity to achieve good results. To explore the week daily patterns we divided the time in three dimensions. Fig. 1 shows the water deposit level signal for five weeks. This sensor gives a percentage of the tank capacity every 5 minutes. When adding new dimensions, an organized structure needs to be conceived to takes advantage of the signal periodicity and achieve good results. To explore both week and day patterns, time is divided in three dimensions. The first ones is the daily hour, and with the 5 minute resolution given by the sensor, it means 288 samples per day. The second one is the week day, from 1 to 7, that is from Monday to Sunday. The third one is the week number relative to the total historical data. A tensor χ of $288 \times 7 \times n_w$ is obtained, where n_w is the week number. It is mandatory to ensure that used data are reliable in order to calculate the MSE per sample. The data blocks with too much long original bursts have to be discarded, so that only 118 of the 150 weeks of data accumulated in the SCADA database can be utilized. As shown in Fig.1, one of the deliberated burst of lost data is remarked in red.Taking into account results achieved in [16], two sensors have been tested: the 3 weeks tensor as the most simple $\chi^{288\times7\times3}$, and the 7 weeks tensor $\chi^{288\times7\times7}$, which already demonstrated achievements on stability. 1a, 1b and 1c portray a graphical example of $\chi^{288\times7\times3}$ for both Non-customized and Customized *tensorization*.

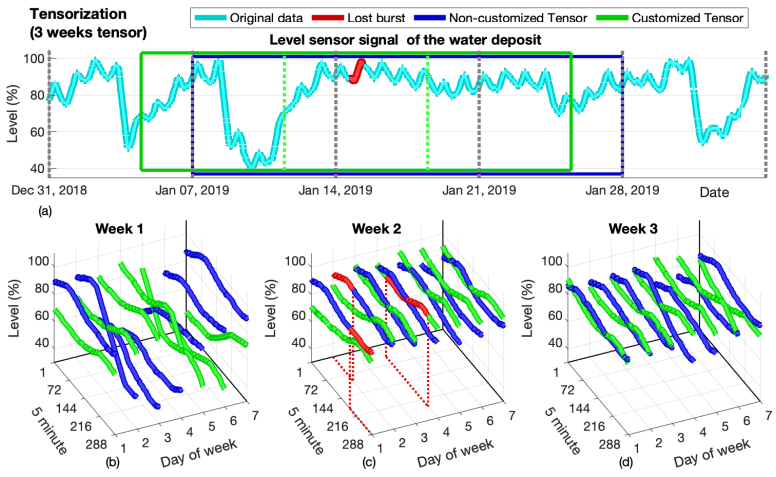

Figure 1. *tensorization* methodology. a) Plot of the of a level sensor signal acquirted for five weeks. Vertical dotted gray lines point out the days of the beginning of the week. The blue window shows the data selected by the Non-customized tensor, to construct a 3 weeks tensor, which includes the week where the burst is, the week before and the week after, to construct the 3 weeks tensor. The green window shows the data selected by the Customized tensor, that allocates the burst is the center. b), c) and d) show the three weeks tensors, in blue the Non-customized one and in green the Customized one. In red is remarked the location of the burst, relative to each tensorThe location of the burst relative to each tensor is in red.

3. Overview on the data completion method used

The operational steps for the reconstruction of the lost bursts of data are here described.

The first step consist in using linear techniques to perform a first restoring of the samples, and to avoid introducing empty data to the tensors.In this study, the *Ramp* and a *Finite Impulse Response (FIR)* methods have been used, as explained in [16].

The *Ramp* method consists in using a simple straight line to connect the last known sample before the burst to the next one after the burst. It is just necessary to calculate the constant increment or decrement, m, and apply this to the whole batch of lost samples, starting from the first sample x_{n+1}, and including x_{n+B}, where B is the burst length. This operation allows to fill the burst without discontinuity.

$$x_{n+i} = x_n + m \cdot i \qquad i \in 1, \cdots, B \qquad \text{where} \qquad m = \left(x_n - x_{n+B+1}\right)/(B+1) \tag{1}$$

The *FIR* method is based on the Wiener Predictor, which finds the optimal coefficients a_k, by utilizing the signal auto-correlation and the criteria of minimum MSE, to make predictions of signal x_n from last received samples. The predictor is re-supplied with each estimated sample until filling the entire lost burst.

$$x_{n+i} = \sum_{k=1}^{K} a_k x_{n+i-k} \tag{2}$$

Where K, is number of coefficients, $\mathbf{R}_x$, the auto-correlation matrix, $\mathbf{r}_x$, the auto-correlation vector, and $a_i = {\mathbf{R}_x}^{-1}\mathbf{r}_x$, the FIR coefficients.The *FIR* method combines the result of two versions of this predictor. The classic one, which uses the historical data samples before the burst, to make the prediction. And a modified version of the same process, using the historical data samples located after the burst. The mean of the two predictions is calculated giving proportionally more weight to the classical prediction at the beginning of the burst, and more weight to the modified version at the last samples of the burst. The problem using only the classic one is that a signal continuity loss often occurs, whilst combining the two predictions the signal continuity is preserved.

In the second step, a block of data including the burst, is *tensorized*. A decomposition technique is used to simplify the data of the container and discovering the relation between dimensions (modes). The Tuker [9] and CP [7,11,10] decompositions are tested and proved. Fig. 2 shows a 3-way tensor representation of the Tuker model applied to $\chi^{I\times J\times K}$. To execute the decomposition it uses a multi-linear transformation with a smaller core $G^{L\times M\times N}$ and the factor matrices $A^{I\times L}$, $B^{J\times M}$ and $C^{K\times N}$ ~~to execute the decomposition~~, that can be written as folows:

$$\chi^{I\times J\times K} \approx G^{L\times M\times N} \times_1 A^{I\times L} \times_2 B^{J\times M} \times_3 C^{K\times N} \tag{3}$$

Where the symbol $\times_i$ stands for the n-way product of a tensor by a matrix as it is defined in [17]. In the Fig. 2b there is a representation of the 3-way CP model applied to $\chi^{I\times J\times K}$. It is a specific case of Tuker decomposition, where L = M = N (= D), with $G^{D\times D\times D}$ diagonal. In terms of the tensor product the CP decomposition can be written as folows:

$$\chi^{I\times J\times K} \approx G^{D\times D\times D} \times_1 A^{I\times D} \times_2 B^{J\times D} \times_3 C^{K\times D} \tag{4}$$

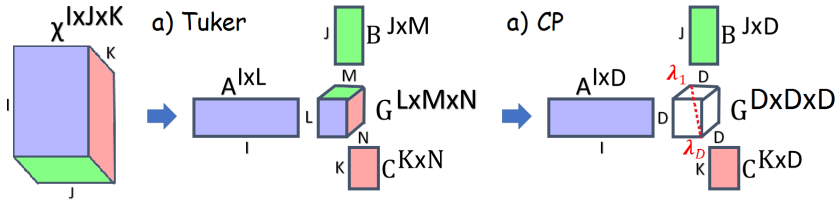

Figure 2. Tensor factorization methods. a) Tucker. b) CANDECOMP/PARAFAC (CP).

The last step is the correction of the offset generated for the tensor decomposition. To force maintaining the signal continuity, as explained in [16], the last received sample before the burst and the first received sample after burst are used. Although the *tensorization* helps to follow the tend of the signal, some spontaneous discontinuities can appear, especially when the burst is located across days or weeks, as observed during this study.

The computational costs, in terms of execution time, for the two algorithms are not excessive. In a 1.000 iterations test, and considering the worse computational conditions of 200 samples of burst length and a 7 weeks tensor, a mean of 141 ms for the Tuker and 147 ms for the CP were obtained. The maximum Tuker decomposition processing time was 402 ms and 363.4 ms for the CP. Simulations were performed with Matlab 2018b installed in a laptop with Windows 7 Professional operating system (Intel(R)571Core(TM) i5-6200U, 2.3 GHz and 8 GB of RAM).

4. A customized *tensorization*. The new tensor ordering proposed

The main contribution of this article is the new way of ordering the data within the tensor. This new method has been developed as a customized *tensorization*. The procedure can be described as follows: the *tensorized* data is re-organized to locate the lost burst in an optimal position. To avoid distortions, the burst is placed in the core of the tensor. As explained in [16], lost data bursts never exceed the day.Their length, n_l, is always less than 288 samples. The tensor $\chi^{I\times J\times K}$ has been considered, setting I=288 and J=7 in order to collect the daily and weekly periodicity. Note that 288 is the number of 5-minute samples collected in one day and 7 the days of a week while W is the number of weeks collected in the tensor.

To construct an appropriate tensor, an odd number (3, 5, 7,...) is used in order to always have a central week in the tensor. Once the burst to be completed is located, its length (n_l) can be determinate as well as its starting and ending time. A central position J=4 and K=0.5(W+1) is selected, so that the initial position of the lost burst is I_i=0.5(288-n_l).The burst samples are placed in the central positions of the tensor $\chi^{I_i:I_i+n_l-1\times 4\times 0.5(W+1)}$. Preserving the temporal order, with the samples before and after the burst, the tensor is filled. Note that proceeding in this way the daily cycles probably do not start at 00:00 and the week do not start on Monday, as occurred in [16]. If the lost samples did not occur on Thursday (that is the pre-selected J=4), the start day of the week must be moved to force the burst day be placed in the center of the week. The new daily cycles will begin $5\times I_i$ minutes before the burst occurs (i.e., if considering 100

samples burst the daily cycle will start at 07:45 AM; or, considering 200 samples length at 03.35AM).These conditions allow to always have the same number of samples, before and after the lost burstas well as achieve an equilibrium between information provided by both past and future samples. The difference between the new and the old *tensorizing* arrangement is observable by comparing the Fig. 5a with the 6a and the Fig. 7a with the 8a, where a 3 weeks (W=3) tensor has been used.

5. Proposed methodology

In this study, real data coming from the water level sensor of the main water tank of AVSA supplier have been used. Firstly, the weeks showing too many original bursts of lost data were discarded, as these are not useful to verify the reliability of the reconstructions. Then, 1000 different starting position have been generated in a random way to simulate the lost bursts, and 100 or 200 consecutive samples were removed, according to the test. Next, the deliberately lost data is restored by performing the best two linear methods used in [16], the *Ramp* and the *FIR*. Finally the linear restorations are *tensorized* to refine the results.

At the beginning, the process was carried without modifying the tensor arrangement, that is by organizing the tensor daily and weekly, starting every day at 00.00 and finishing at 24:00, and starting every week on Monday, independently of the burst position. Using this process, the consistency of the methodology in [16] was confirmed. However, as described in the next section, in some cases, a discontinuity appears and damages the reconstruction of the signal, mainly when the burst is located between two different weeks or days as shown in Fig. 7. To avoid this, another tensor configuration was tested, based on the burst position, and forcing the burst to be on the center of the tensor. With these new tensor arrangement, it has been possible to reproduce the experiment to confirm statistically the improvements reached.

6. Results

In this section, completion algorithms with and without tensor rearrangement have been compared by randomly erasing continuous data streams of length 100 and 200. As in [16] MSE per sample is used. As for the experiments carried on, two sizes of tensors $\chi^{288\times7\times W}$ were considered, that is W=3 and W=7, corresponding to 3 and 7 weeks. To evaluate the MSE per sample, 1.000 iterations for each algorithm configuration are performed to ensure statistical significance.

The method initially proposed is based on a first step that uses linear methods to approximate the solution. For this first stage, two options are tested, the *Ramp*, and the more sophisticated *FIR* predictors. This phase allows to successively perform an estimation of the data that are useful to *tensorize* avoiding empty elements.

In the second stage the tensor is decomposed according to the best options resulted in [16]: for the Tucker, the decompositions: TK(1,1,1), TK(2,2,1), TK(3,3,1), TK(2,2,2) and TK(3,3,3). For the CAN-DECOMP/PARAFAC, de options CP(2) and CP(6). Fig. 3 shows the performance (MSE per sample) of the different original algorithm configurations used. In Fig. 4 the same algorithm options are tested when the *tensorization* pro-

cess is performed according to the simulation test parameters above described. Note the remarkable improvement of the MSE per sample obtained in that second case. Fig. 5 and 6 show the results of the *tensorization* and recovering of a 200 samples burst, using both Non-customized and Customized algorithms. In both cases the tensor packages 3 weeks of data and the decomposition used is the TK(3,3,1), which shows the best results in all cases. Fig.s 7 and 8 show another practical example carried with the same conditions and showing discontinuity effects that arose with the starting algorithm, but not with the customized one, which was not affected.

7. Conclusions

It has been proved that the implementation of an appropriate tensor arrangement is of a great importance for achieve more benefit from the signal periodicity that helps the data reconstruction. As depicted in Fig. 4, the Customized tensor shows a better performance with a MSE decrease, indifferently of the burst length and the tensors time lapse in weeks, as well as of the tensor configuration. It is worth to note that the same percentage of improvement of 13% occurs using both the Non-customized and the Customized tensor, therefore showing a proportional error decrease. As for this study, the Tuker strategy has been proved the be best type of tensor decomposition, and TK(3,3,1) as its optimal configuration.

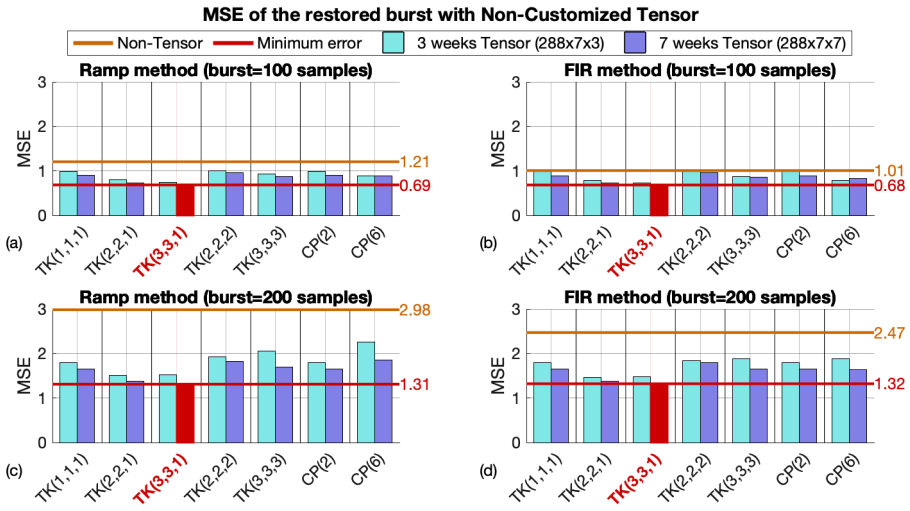

Figure 3. MSE of the Non-Customized Tensor by a 1000 iterations simulation. Two tensors are tested, with 3 and 7 weeks. Two linear methods are applied as the first restoration stage, *Ramp* and *FIR*. Two burst lengths are considered, 100 and 200 samples. The comparison between deleted original data and reconstructed data is performed by calculating the MSE per sample.

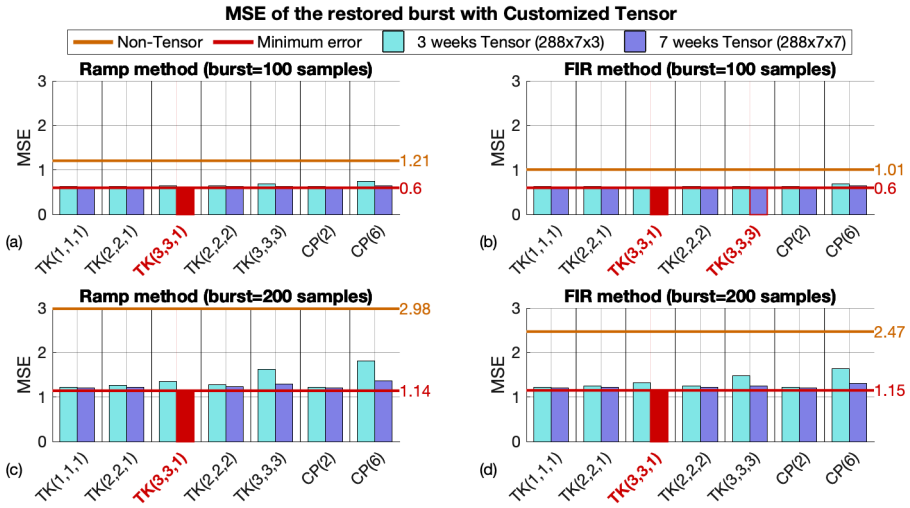

Figure 4. MSE of the Customized Tensor by a 1000 iterations simulation. Two tensors are tested, with 3 and 7 weeks. Two linear methods are applied as the first restoration stage, *Ramp* and *FIR*. Two burst lengths are considered, 100 and 200 samples. The comparison between deleted original data and reconstructed data is performed by calculating the MSE per sample.

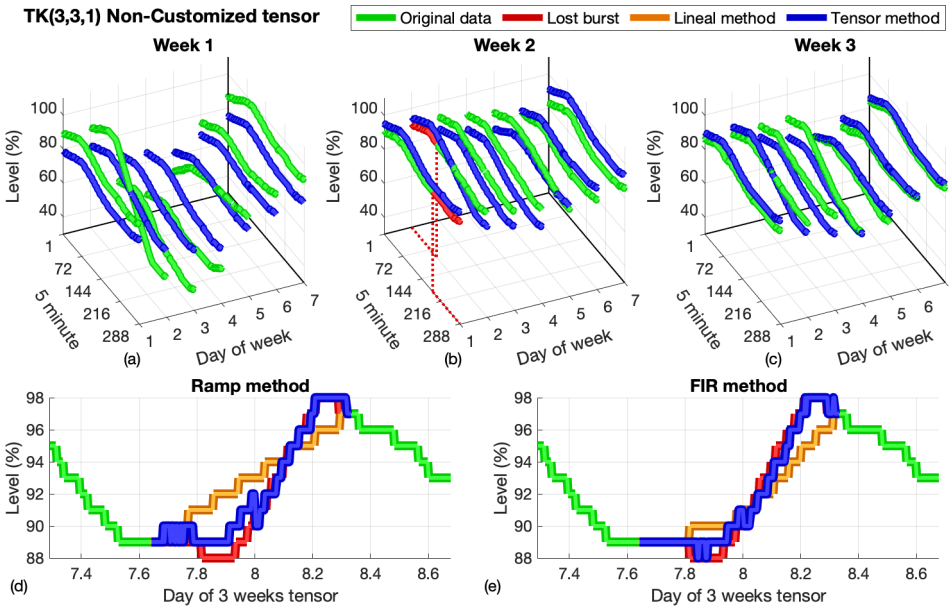

Figure 5. Non-customized tensor restoration with a 200 samples burst wholly located in the same week. Tensor with three weeks of data (288,7,3). (a),(b) and (c) show the tensor and the red line remarks the burst. (d) and (e) show the two stages, linear and tensor, of the restoring process.

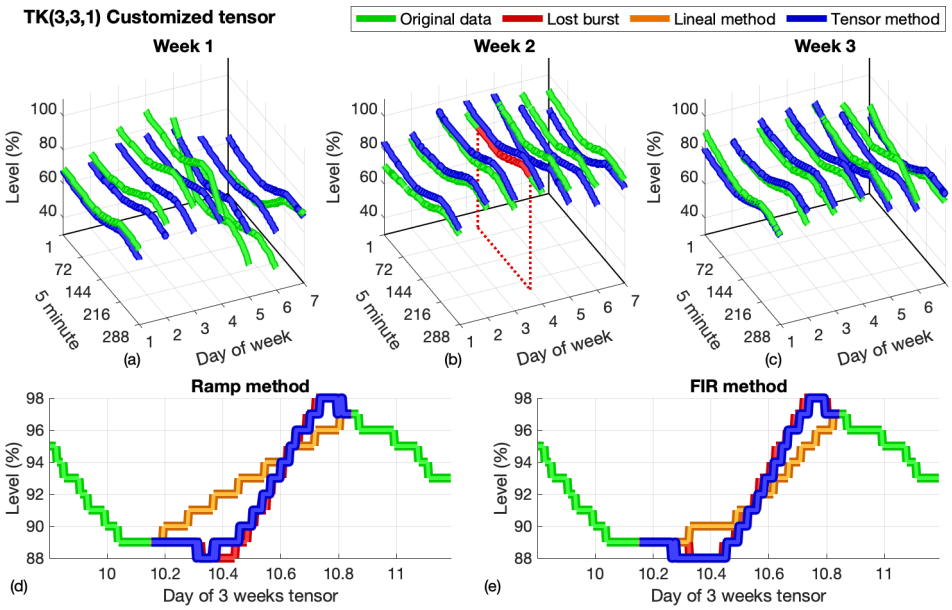

Figure 6. Customized tensor restoration with a 200 samples burst wholly located in the same week. Tensor with three weeks of data (288,7,3). (a),(b) and (c) show the tensor and the red line remarks the burst. (d) and (e) show the two stages, linear and tensor, of the restoring process.

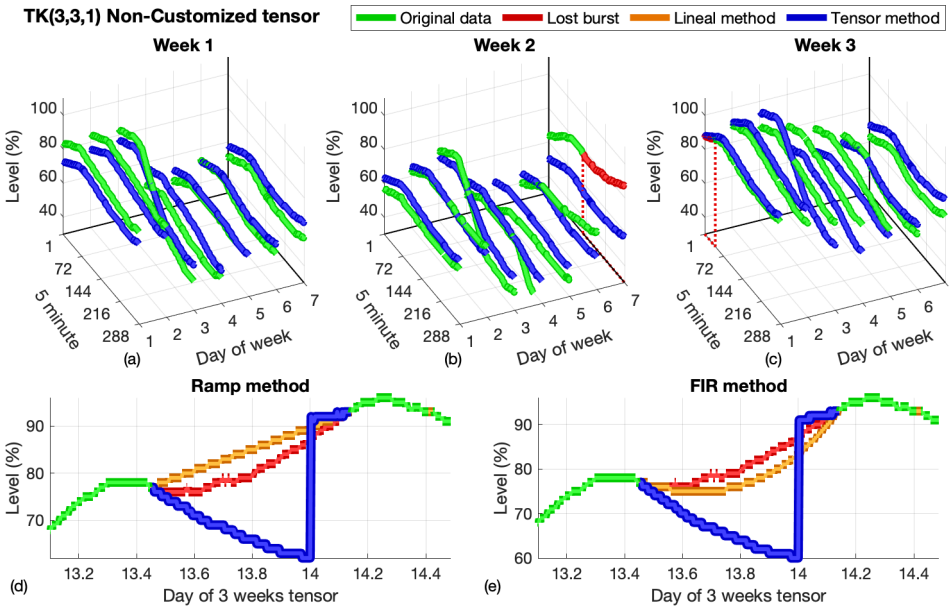

Figure 7. Non-customized tensor restoration with a 200 samples burst located among two weeks, and where the discontinuity distortion appeared. Tensor with three weeks of data (288,7,3). (a),(b) and (c) show the tensor and the red line remarks the burst. (d) and (e) show the two stages, linear and tensor, of the restoring process.

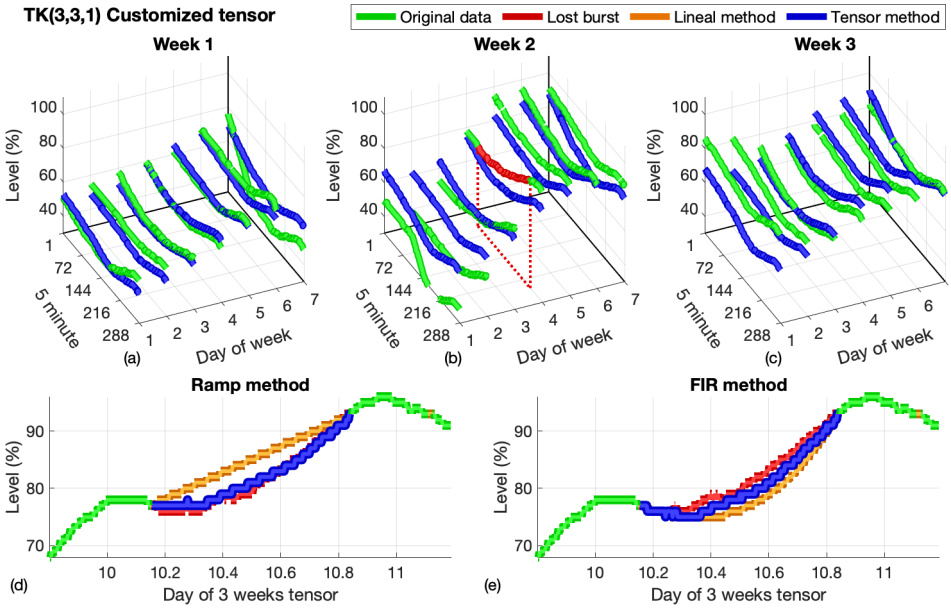

Figure 8. Customized tensor restoration with a 200 samples burst located among two weeks, and where the discontinuity distortion is prevented. Tensor with three weeks of data (288,7,3). (a),(b) and (c) show the tensor and the red line remarks the burst. (d) and (e) show the two stages, linear and tensor, of the restoring process.

Acknowledgements

Financial support by the Agency for Management of University and Research Grants (AGAUR) of the Catalan Government to Arnau Martí-Sarri is gratefully acknowledged. We thank the company Aigües de Vic S.A. for giving us access to their databases. We especially thank Mainardo Gaudenzi Asinelli, a PhD of the UVIC, for his help with the revision of English.

References

[1] B. Lamrini, E.-K. Lakhal, M.-V. Le Lann and L. Wehenkel, Data validation and missing data reconstruction using self-organizing map for water treatment, *Neural Computing and Applications* **20**(4) (2011), 575–588.

[2] J. Blanch, V. Puig, J. Saludes and J. Quevedo, Arima models for data consistency of flowmeters in water distribution networks, *IFAC Proceedings Volumes* **42**(8) (2009), 480–485.

[3] V. Puig, C. Ocampo-Martinez, R. Pérez, G. Cembrano, J. Quevedo and T. Escobet, *Real-Time Monitoring and Operational Control of Drinking-Water Systems*, Springer, 2017.

[4] M.À. Cugueró-Escofet, D. García, J. Quevedo, V. Puig, S. Espin and J. Roquet, A methodology and a software tool for sensor data validation/reconstruction: Application to the Catalonia regional water network, *Control Engineering Practice* **49** (2016), 159–172.

[5] E. Acar, D.M. Dunlavy, T.G. Kolda and M. Mørup, Scalable tensor factorizations for incomplete data, *Chemometrics and Intelligent Laboratory Systems* **106**(1) (2011), 41–56.

[6] M. Signoretto, R. Van de Plas, B. De Moor and J.A. Suykens, Tensor versus matrix completion: a comparison with application to spectral data, *IEEE Signal Processing Letters* **18**(7) (2011), 403.

[7] R.A. Harshman, Foundations of the PARAFAC procedure: Models and conditions for an" explanatory" multimodal factor analysis, *Working Papers in Phonetics* **UCLA Working Papers in Phonetics, 16** (1970), 1–84.

[8] M. Sørensen, L.D. Lathauwer, P. Comon, S. Icart and L. Deneire, Canonical polyadic decomposition with a columnwise orthonormal factor matrix, *SIAM Journal on Matrix Analysis and Applications* **33**(4) (2012), 1190–1213.

[9] L.R. Tucker, Some mathematical notes on three-mode factor analysis, *Psychometrika* **31**(3) (1966), 279–311.

[10] J.D. Carroll and J.-J. Chang, Analysis of individual differences in multidimensional scaling via an N-way generalization of Eckart-Young decomposition, *Psychometrika* **35**(3) (1970), 283–319.

[11] L. De Lathauwer, B. De Moor and J. Vandewalle, A multilinear singular value decomposition, *SIAM journal on Matrix Analysis and Applications* **21**(4) (2000), 1253–1278.

[12] M. Mørup, Applications of tensor (multiway array) factorizations and decompositions in data mining, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **1**(1) (2011), 24–40.

[13] T.G. Kolda and B.W. Bader, Tensor decompositions and applications, *SIAM review* **51**(3) (2009), 455–500.

[14] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa and H.A. Phan, Tensor decompositions for signal processing applications: From two-way to multiway component analysis, *IEEE Signal Processing Magazine* **32**(2) (2015), 145–163.

[15] P. Comon, Tensors: a brief introduction, *IEEE Signal Processing Magazine* **31**(3) (2014), 44–53.

[16] P. Marti-Puig, A. Martí-Sarri and M. Serra-Serra, Different Approaches to SCADA Data Completion in Water Networks, *Water* **11**(5) (2019), 1023.

[17] T.G. Kolda, Multilinear operators for higher-order decompositions., Technical Report, Sandia National Laboratories, 2006.

Apèndix D

DOUBLE TENSOR-DECOMPOSITION FOR SCADA DATA COMPLETION IN WATER NETWORKS

MDPI

Article

Double Tensor-Decomposition for SCADA Data Completion in Water Networks

Pere Marti-Puig [1,‡*], Arnau Martí-Sarri [13,‡] and Moisès Serra-Serra [2,]

[1] Data and Signal Processing Group, U Science Tech, University of Vic - Central University of Catalonia, c/ de la Laura 13, 08500 Vic, Catalonia, Spain
[2] MECAMAT Group, U Science Tech, University of Vic - Central University of Catalonia, c/ de la Laura 13, 08500 Vic, Catalonia, Spain
[3] Aigues de Vic S.A., c/ Santiago Ramon y Cajal 60, 08500 Vic, Catalonia, Spain
* Correspondence: pere.marti@uvic.cat; Tel.: +34-93-881-55-19
‡ These authors contributed equally to this work.

Version October 27, 2019 submitted to Water

Abstract: Control And Data Acquisition (SCADA) systems currently monitor and collect a huge among of data from all kind of processes. In practice, due to sensor failures or to communication errors, in the long-time running, some data may be lost. When it happens, given the nature of these failures, information is lost in bursts, that is, sets of consecutive samples, which besides can be very long. Data completion is a critical step, which must be done with the utmost rigour in order to not propagate errors in the rest of the processing chain stages. Some Big Data techniques do not work if the data series are incomplete, due to the loss of some data. When this occurs it is necessary to fill out the gaps of the historical data with a reliable data completion method. This paper presents an *ad hoc* method to completion the data lost by a SCADA system in case of long bursts. The data correspond to levels of drinking water tanks of a Water Network company that present patterns on a daily and a weekly scale. A method based on tensors is used to take advantage of the data structure. A specially designed *tensorization* is employed to deal with bursts of missed data, applying a twice tensor decomposition and a signal *continuity correction*. Statistical tests are realized, which consist of apply the data reconstruction algorithms, by deliberately removing bursts of data in verified historical database servers, to be able to evaluate the real effectiveness of the tested methods. For this application, the presented approach outperforms the other techniques found in the literature.

Keywords: Water Networks; SCADA Data; Tensor completion; Tensor decomposition

1. Introduction

Currently, the data collection has made a real breakthrough with the many variety of sensors and devices which have the possibility of transmit information from anywhere. With the increase of the data storage capacity in the world of computers, the point of save more data than it can be treated is reached.

In practice, when processing this amount of information, the problem of incomplete or missing data has to be addressed. The Data management in water networks [1] and in hydrological resources [2–4] are not an exception. That problem is especially challenging when it manifests itself in long bursts. Aigues de Vic S. A. (AVSA) decided three years ago to renew their Supervisory Control And Data Acquisition (SCADA) system, because it was been becoming outdated. AVSA is the enterprise responsible for the water supply of the city of Vic. The SCADA is a tool for the technicians of the Water Purification Plant (WPP), where the Ter river water is purified, and for the operators of the Water Distribution System (WDS). The old system is usefully to receive information of the sensors and take

decisions, but not to remotely configure and control the devices. For example in the case of a pumping system, it is possible to see the pumps configuration, but if it is necessary to reduce the pumped water flow, the operator have go where the pumps are located and do it manually.

Something important on the SCADA system renovation is to take advantage of the data collected by the old SCADA. To avoid the lost of information accumulated by the old SCADA system during the last four years, the most important data have to be imported from the old data base to the new one. During this duty the historical data series were verified with the aim to not import unusable data, and some problems related to missing data were detected. An example of this was the case of the data collected by the deposit level sensor located in the main water reservoir of the city of Vic. It is important to preserve this data, because the historical data of this sensor could be used, for example, to find patterns on the city of Vic consumption. To restore lost samples some simple linear estimators with acceptable results were used [5–7], but in the case of large amounts of consecutive lost samples, the linear estimators lost their effectiveness. These type of lost data is caused basically by a fail in the communication between the Programmable Logic Controller (PLC), where the sensor is connected, and the central SCADA server, where the data is stored.

The classical methods of data estimation hardly exploit simple patterns, which can appear daily, weekly or in general with a concrete frequency. In contrast, the methods based on tensor decomposition are able to take advantage, with a multidimensional way, of the appropriately arranged data [8–13].

In a previous study [14], an *ad hoc* method was implemented using tensors, which is specially adapted to work with water deposit level signals and to deal with long bursts of lost samples. The method was compared with other reconstruction method based on tensor procedure found in the literature, giving better results for this specific conditions. Since the signals of interest present daily and weekly patterns, the approach in [14] combines classical interpolation strategies with techniques of tensors decomposition and a *continuity correction method* that guaranteed the continuity of the signal of the data recovered.

The method presented in this paper is reminiscent of [14] in some aspects, but presents some novelties that significantly impact in the performance. The method starts by filling the lost burst values to avoid missing elements before to *tensorizing* the data. At this point, two significant differences are introduced: (1) to fill empty values the most straightforward interpolation is chosen, which is the called *ramp method*, discarding the predictive or the extremely simple methods proposed in [5–7] or in [14], (2) the way to organize the tensor is imporved by introducing what we call the *burst centered tensorization*. The most significant difference, however, is that the new method employs the reconstruction methodology twice, using two different *tensorization* cores in the tensor decomposition step. The first one, perform the tensor decomposition with a small-dimension *tensorization* core, obtaining a first approximation. The second one uses a large-dimension *tensorization* core in the tensor decomposition procedure, what allows to refine the first estimation. Note that, although several tensor decompositions exist, the two most extended and well-known are the Tucker [15–17] and the CANDECOMP/PARAFAC (CP) [18,19] which are the two decompositions considered in this work as well as in [14]. References [11–13,20] can provide to the reader a quality tensor algebra introduction. For the type of signals treated, when data losses are distributed uniformly or even in short bursts of less than 30-40 samples, all methods work more or less likewise. Above that length, tensor-based methods begin to take advantage. In practice, it is observed that the bursts length of data lost on a SCADA system communication cutting off can be longer than this. The proposal obeys the need to improve the performance of the data replenish methods currently used. The main contribution of this research is to improve the data reconstruction methodology developed in [14], whose results are taken as a reference, since they were better in comparison with the proven tensor methods that already exist in the literature.

Henceforth, the work is organized as follows. In the *Materials and methods* section the details for reproducing results are explained. Aspects related to the database and its pre-processing are treated briefly because they are the same as those carried out in [14]. The same is applied for tensor concepts.

The focus is on the process *burst centered tensorization* and on the *double decomposition* of the tensor. Although a *smoothing data process* is applied before the *tensorization*, which contributes to achieve better results. In the *Results* section, the methodology is tested applying only each of the proposed improvements and applying them all together, in order to quantify the impact of each of them and to check if they are complementary. Finally, the most remarkable aspects will be summarized in the *Discussion and Conclusions* section.

2. Materials and Methods

2.1. Used database

The historical data used to perform the simulations are provided by Aigues de Vic S.A. (AVSA). Their Supervisory Control And Data Acquisition (SCADA) system collect approximately 1,300 different signals. Specifically, the data used on the simulations is provided by a water level sensor located in the deposit of Castell d'en Planes, which is the water reserve of the city of Vic. The data of this sensor was collected from 1 October 2015, but some weeks of the historical data have to be discarded for the simulations. The data used for the simulations is verified, discarding the weeks where there are excessive lost data, because not allow to calculate the real MSE and verify the results.

2.2. Imputation method: the ramp method

The tensor decompositions cannot work with empty data. One of the simplest strategies used with acceptable results in [14], called the *ramp method* is used in this study. The *ramp method* consists of filling the lost data by drawing a line between the last known sample before the lost burst starting, x_n, and the first known sample after the lost burst ending, x_{n+B+1}, where B is the length of the data burst lost in number of samples. So that, considering a lost burst of B samples and the index i going from 1 to B, to use a constant increment (or decrement), m, and fill the entire lost burst, x_{n+i} must be:

$$x_{n+i} = x_n + m \cdot i \quad \text{for} \quad m = \frac{x_n - x_{n+B+1}}{B+1}. \tag{1}$$

Fig.1 shows the performance of the *ramp method*.

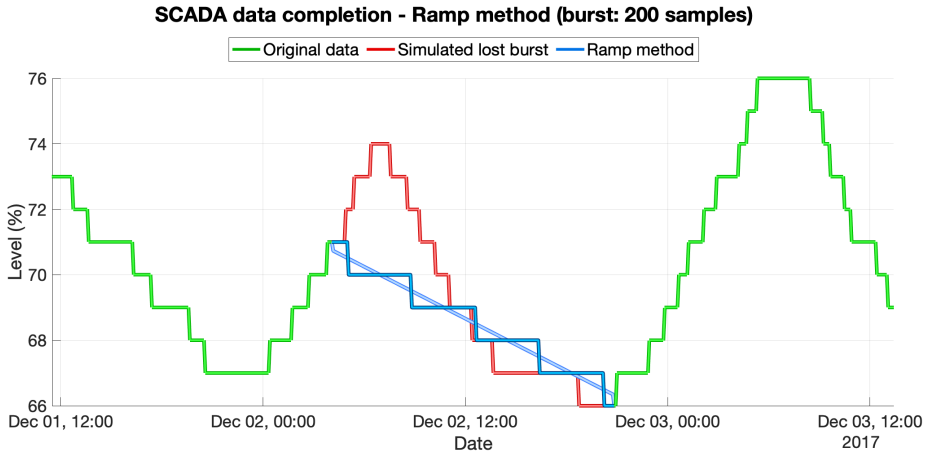

Figure 1. First step of the data reconstruction method. The red line shows the burst of the simulated lost data. The blue line corresponds to the data reconstruction of the linear method called the *ramp method*. The soft blue line shows the linear method result and the strong blue line shows the final signal reconstruction, which is adapted to the sensor resolution of 1%.

2.3. Burst centered tensorization

Tensorization is the process of packaging lower-dimensional data into a container, the tensor, with more dimensions than the original one. This allows us to find the relations between dimensions, which are difficult to perceive in more simple structures. The visual inspection of the data seems to reveal patterns on a daily and weekly scale. To take advantage of these regularities, a 3-dimensional tensor was composed. The first index indicates the 5-minute day-intervals fixed by the sample frequency of the SCADA system (so each day is represented by 288 samples). The second index indicates the day of the week (which is an index of 7 positions, corresponding to the days of the week). Finally, the third index depends on the number of weeks included in the tensor n_w. In this way, the 3-dimensional tensor will have the following $288 \times 7 \times n_w$ structure.

The proposed organization uses past and future data with respect to the burst location in order to contribute with past and future information. Typically, the way to organize the data into a tensor does not take into account the position of the lost data. In this work, a *burst centered tensorization method* is proposed, where the data selected to fill the tensor container depends on the burst location. Fig.2 shows this process. In Fig.2(a), the dark blue window shows the selection of data as was proposed in [14], in a typical way and with the data presented in a uni-dimensional view. The week where the burst is located is used as the central week, and some weeks before and some weeks after are taken, depending on the tensor size n_w. Note that to always have a central week in the tensor n_w there must be an odd value (3, 5, 7, ...). Through the dark blue window it can be seen how the burst is not exactly located in the center of the selected data, which would mean being in the center of the dark blue window, even if the week where the burst is located is selected as the central week. This happens because the burst is hardly located in the middle of a week, which only happens if the burst is located exactly in the center of Thursday. The *burst centered tensorization* forces the burst to be located in the center of the data selected. The cyan window in Fig.2(a) shows this new data selection, where the burst is placed exactly on the center of the window. Fig.2(b), (c) and (d) show the *tensorized* data by the typical way. The Fig.2(e), (f) and (g) show the *tensorized* data from the *burst centered tensorization* method, which placed the burst in the core of the tensor (in the center of the central day of the week, which is located in the middle of the tensor). As explained in [14], lost data bursts never exceed the day, meaning that their length, B, is always less than 288 samples and that the burst can be located in the center of a day. Thus, given a B burst in a tensor $\chi^{I \times J \times K}$, the burst samples are placed at $J = 4$, $K = 0.5(W + 1)$ with initial position I_i=0.5(288-B) according to and indexation $\chi^{I_i:I_i+B-1 \times 4 \times 0.5(W+1)}$. Note that the daily cycles of the *burst centered tensorization* rarely start at 00:00 and the weeks do not start on Mondays, as occurred in [14], however there are always the same number of samples before and after the lost burst, which hardly happens with the previous *tensorization* method.

2.4. Tucker and CP tensor decompositions overview

A tensor is a container that can arrange data in N-ways or dimensions. An N-way tensor of real elements is denoted as $\chi \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and its elements as: $x_{i_1,i_2,\ldots,i_N}$. According this, an $N \times 1$ vector $\mathbf{x}$ is considered a tensor of order one, and an $N \times M$ matrix $\mathbf{X}$ (or $\mathbf{X}^{N \times M}$), a tensor of order two.

The procedure of reshaping a lower-dimensional original data (for instance a vector or a matrix) into a tensor is referred to as *tensorization*. The process of reshaping tensors to vectors is named *vectorization*.

Low order tensor decompositions provide a simplified version of the data while making the relation between dimensions explicit. In the case of the 3-dimensional tensor $\chi^{I \times J \times K}$ the approximations are given in the form of a smaller tensor core $G^{L \times M \times N}$, (where $I > L$, $J > M$, and $K > N$) and the L, J and K eigenvectors of mode-1, -2, and -3 respectively which are organized as column vectors in matrices $\boldsymbol{A}^{I \times L}$, $\boldsymbol{B}^{J \times M}$, and $\boldsymbol{C}^{K \times N}$. The size ($L$,$M$,$N$) of the core determines the level of the decomposition.

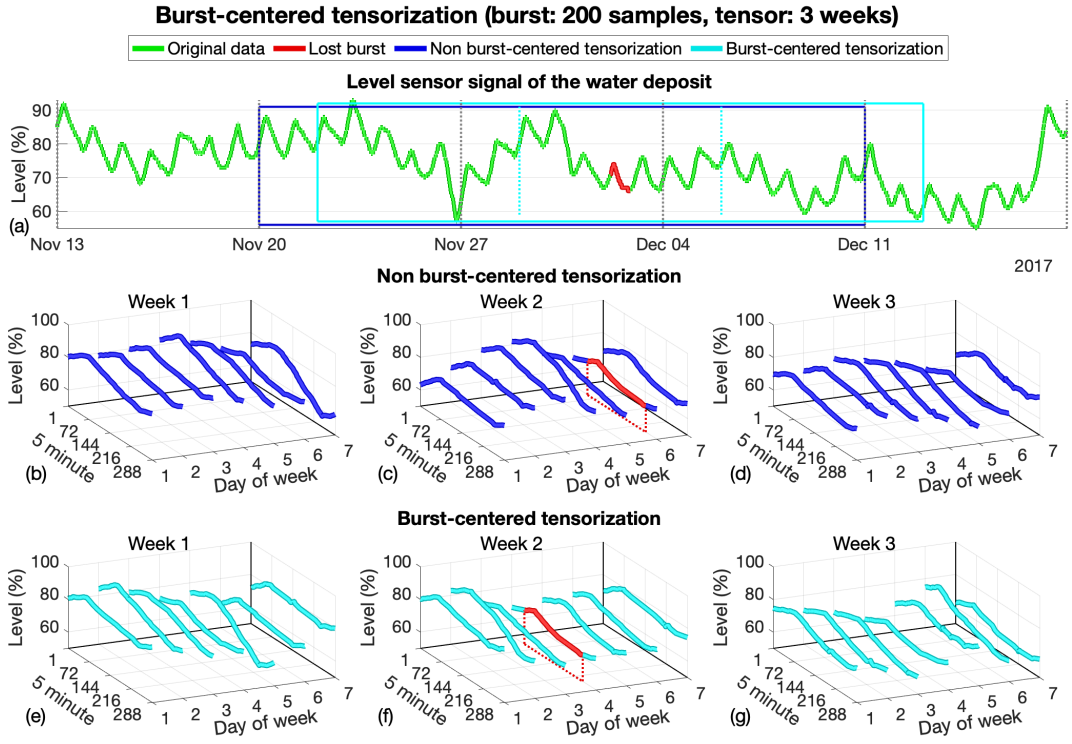

Figure 2. Example of the data *tensorization* of a 3 week tensor with 200 samples of data burst lost. In figure a) the green line shows the original data and the red line shows the lost burst. The strong blue window shows the data introduced in the *non burst-centered* tensor and the soft blue one shows the data introduced in *burst-centered* tensor, which forces the burst to be on the center of the window. Figures b), c), and d) show the three weeks of the *non burst-centered* tensor and the location of the burst, which is located on the central week but not on the center of this week. The figures e), f), and g) show the 3 weeks of the *burst-centered* tensor and the new location of the burst in the core of the tensor, which is in the middle of the central week.

There are many known tensor decompositions but overall of them the most widely used are the Tucker [15] and the CP [18] decompositions. In this work we only test those two. However, the method presented can be adapted to work with any one of them. These two are briefly presented below.

In the 3-way Tucker decomposition model the core is defined by parameters L, M, N, relative to the size of $G^{L\times M\times N}$ and it is expressed as Tucker(L,M,N) according to:

$$\chi^{I\times J\times K} \approx G^{L\times M\times N} \times_1 \boldsymbol{A}^{I\times L} \times_2 \boldsymbol{B}^{J\times M} \times_3 \boldsymbol{C}^{K\times N}, \tag{2}$$

where the symbol $\times_i$ is the n-way product of a tensor by a matrix; such a tensor operation defined, for example, in [21].

The 3-way CANDECOMP/PARAFAC (from CANonical DECOMPosition/PARAllel FACtorization) model is commonly known as CP and can be seen as particular case of the Tucker decomposition when $G^{D\times D\times D}$ is diagonal. Taking this observation into account the CP decomposition can be written in the same terms as in the case of Tucker decomposition, as follows:

$$\chi^{I\times J\times K} \approx G^{D\times D\times D} \times_1 \boldsymbol{A}^{I\times D} \times_2 \boldsymbol{B}^{J\times D} \times_3 \boldsymbol{C}^{K\times D} \tag{3}$$

although, being $G^{D\times D\times D}$ diagonal, it is frequent to see it written in function of the elements λ_i of the diagonal such as:

$$\chi^{I\times J\times K} \approx \sum_{i=1}^{D} \lambda_i \mathbf{a}_i \circ \mathbf{b}_i \circ \mathbf{c}_i, \tag{4}$$

where the symbol $\circ$ stands for the outer product and the column vectors $\mathbf{a}_i$, $\mathbf{b}_i$ and $\mathbf{c}_i$ are related with the matrices of equ. (3) according to: $\boldsymbol{A}^{I\times D} = [\mathbf{a}_1 \cdots \mathbf{a}_D]$, $\boldsymbol{B}^{J\times D} = [\mathbf{b}_1 \cdots \mathbf{b}_D]$ and $\boldsymbol{C}^{K\times D} = [\mathbf{c}_1 \cdots \mathbf{c}_D]$.

The algebra of tensors is explained in detail and often with the support of graphical illustrations in [10–12,20]. Fig. 3 shows a unified representation of both 3D tensor decompositions.

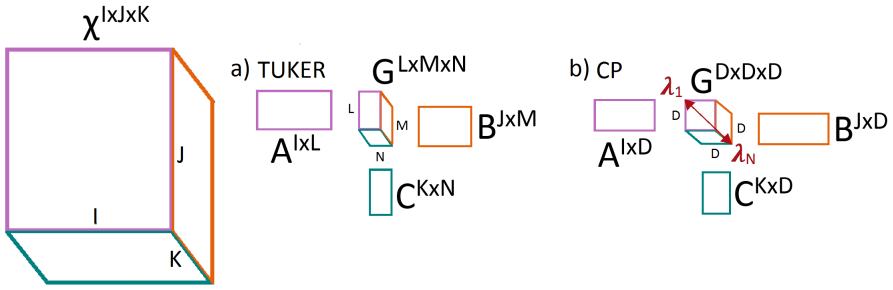

Figure 3. Diagram of the *Tensorization* methods. a) Tucker model. b) CANDECOMP/PARAFAC (CP) model.

2.5. The continuity correction

This procedure was developed in order to maintain the continuity of the estimate provided by a tensor decomposition in its vector form $\hat{\mathbf{x}}$ and the known values of $\mathbf{x}$ at the edges of the burst. As consistently observed previously, the samples in the burst positions after a low-rank tensor reconstruction follow the original signal pretty well but with significant discontinuities in the extremes. Considering x_0 to be the last original known sample before the burst and $\hat{x}_0$ the sample from the tensor reconstruction in that position, we define the initial burst offset as $O_0 = x_0 - \hat{x}_0$. Similarly, for a lost burst of length B, the final burst offset can be defined as: $O_{B+1} = x_{B+1} - \hat{x}_{B+1}$. The corrected offset estimates $\tilde{x}_i$ are computed as follows:

$$\tilde{x}_i = \hat{x}_i + \frac{(B-i)O_0 + (i-1)O_{B+1}}{B-1} \qquad i = 1, \cdots, B. \tag{5}$$

172 Fig. 4 shows graphically the *continuity correction* applied.

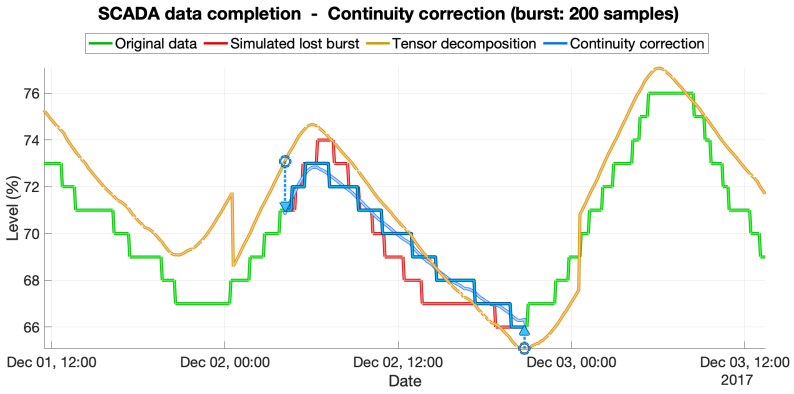

Figure 4. Second and third steps of the data reconstruction method. The green line is the original data, x_i. The red line shows the 200 samples burst of simulated lost data. The Tucker decomposition (1,1,1) is used. The orange line is the result of the tensor process with this configuration, $\hat{x}_i$. The blue arrows indicate the initial and the final offsets, O_0 and O_{B+1}. The soft blue line show the effect of the *continuity correction*, $\tilde{x}_i$ and the strong blue line shows the final signal reconstruction, which is adapted to the sensor resolution of 1%.

173 ### *2.6. Signal smoothing*

174 The tensor decomposition produces a continuous response. The sensor, however, measures
175 the level as a percentage with resolution of 1%, providing a discrete signal of the deposit capacity.
176 When the signal levels oscillate around the point of quantification, oscillations occur between adjacent
177 discrete values. The goal of this section is to verify if a smoothing of the data applied before the tensor
178 decomposition can help to improve the results.

179 The smoothing algorithm adopted is *ad hoc*, developed considering the sensor way of working.
180 The samples are processed in groups with the same integer value, and taking into account whether the
181 signal is increasing, decreasing or is in a relative minimum or maximum. The blocks of samples of
182 identical integer value A are processed taking into account the values of the contiguous blocks. The
183 procedure is effortless. There are more elaborate filtering methods but those introduce delays in the
184 signal, and thus of that this straightforward solution has been chosen instead. If the block corresponds
185 to a signal increment, a line with a positive slope is built with extreme values A-0.5 and A+0.5. If
186 the block corresponds to a signal decrement, a line with a negative slope is built similarly. If it is
187 detected that it is a local maximum or minimum, the block is replaced by a triangular shape with the
188 corresponding orientation. Fig. 5 shows the smoothing performed through an example.

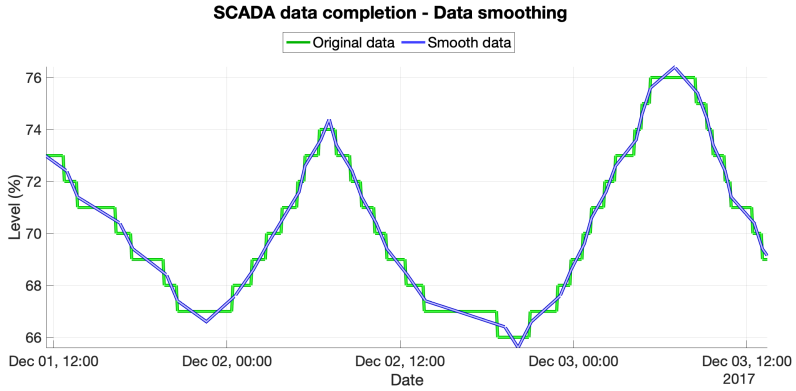

Figure 5. Smooth process applied to the level sensor signal before the second step of the methodology, the *tensorization* of the data, to help on the tensor process to achieve a better estimation.

2.7. Double decomposition approach

In this section the proposed data completion method is presented through the sub-processes previously commented. As was already mentioned, only the two more widely known tensor decomposition models have been considered, these being the Tucker and the CP. Thus the configuration of both decomposition algorithms have been analyzed with the aim of taking the biggest advantage possible from each one.

In order to clarify the followed procedure, it is shown through a particular case in Figure 6, where the steps of the process are represented by using a block diagram.

In that figure, vector $\boldsymbol{x}$ is the input that contains the burst of missing values. The first step required is to choose the linear method of data completion that fills the data gap of the lost burst with a first simple first estimation. The *ramp method* is the selected one, which draws a line to join the known extreme values that delimit the burst as explained in subsection 2.2. It is a rough approximation, but does not need any configuration, which brings simplicity to the algorithm.

Once the data in $\boldsymbol{x}$ has no empty values, and the positions of lost burst values have been saved, the *burst centered tensorization* explained in section 2.3 is applied. The tensor obtained is $\chi^{288\times7\times n_w}$. To remember, its first dimension indexes the SCADA measurements taken in 24h every 5-minute, its second dimension indexes the 7 days to complete a week, and its third dimension indexes the number n_w of weeks considered (which is an odd number: 3 or 7 in the tests realized). In Figure. 6 the particular case of $n_w = 7$ is shown.

This is followed by the first tensor decomposition. The goal is to build a low-range approach of $\chi^{288\times7\times n_w}$ that is done by decomposing Tucker (4,6,1). The result is named $\chi_{(1)}^{288\times7\times n_w}$. At that time, the *continuity correction* is applied on the set of estimated samples which are in the position of the lost burst, with the aim to adapt the estimated signal fluctuation to the original data. That is, to use this set of samples to fill the same positions (those of the lost burst) in a new tensor, called $\chi'^{288\ times7\times n_w}_{(1)}$, which has the rest of positions filled with the original data. Note that both $\chi_{(1)}^{288\times7\times n_w}$ and $\chi'^{288\times7\times n_w}_{(1)}$ have the same tensor arrangement, the *burst centered tensorization*.

The next step consists in doing the second tensor decomposition, now with the second core selected (Tucker(4,7,7) in the example) to obtain an approximation of $\chi'^{288\times7\times n_w}_{(1)}$ named $\chi_{(2)}^{288\times7\times n_w}$. Again, the set of samples of $\chi_{(2)}^{288\times7\times n_w}$ located in the positions of the lost burst are taken to apply the *continuity correction* with the original data, and finally obtaining the SCADA data completion.

The Tucker(4,6,1) and Tucker(4,7,7) decomposition shown in Figure 6(a) correspond to the values that optimize in our database the recovery of a burst of 200 lost samples when employing the *tensorization* of size $288 \times 7 \times 7$ and the Tucker decomposition is used. The method is the same for the CP decomposition. The decomposition CP(1) and CP(15) shown in 6(b) to optimize the recovery

of a burst of 200 lost samples using the *tensorization* of size $288 \times 7 \times 7$ and the CP decomposition. The results are a statistical measure obtained after running 1000 simulations.

The study to determine the optimal size of these decompositions can be found later in subsection 3.1.2.

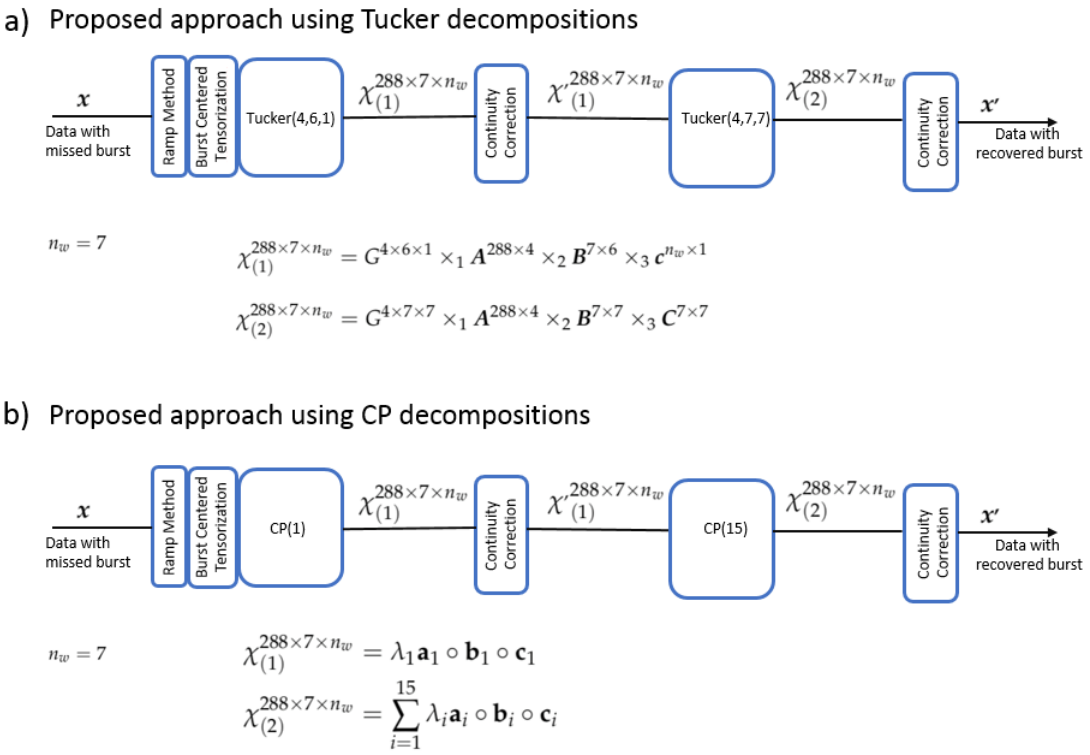

Figure 6. Graphic representation of the *double decomposition method* for both models checked, Tucker (a) and CP (b). In both cases the order of the decomposition shown is the optimized one in order to recover a burst of 200 lost samples when employing a tensor size of $288 \times 7 \times 7$.

2.8. Algorithm performance evaluation

To test all the methods on the same conditions, firstly 1.000 different starting positions are randomly selected from the 77 weeks of historical data previously verified. These set of starting positions determine the groups of consecutive samples which are deleted to simulate the burst of missing data. The strategy, the data set, the block of 77 consecutive weeks, and the burst lengths B were the same as used in [14] in order to compare the evolution of the algorithm performances. When an algorithm replenishes the missing burst, the Mean Square Error (MSE) per sample with the original data is computed. The same algorithm processes those 1.000 different randomly selected cases and the MSE per sample is taken as the parameter to evaluate its performance. Before calculating the MSE, the reconstructed signal is adapted to the sensor resolution of 1% by rounding the values with decimals to the nearest integer the values with decimals. Then, considering $\hat{x}_i$ to be the samples provided by a completion algorithm and x_i to be the true values that had been eliminated in the verified data set to simulate a lost burst of length B, the MSE per sample is computed as:

$$MSE = \frac{1}{B} \sum_{i=1}^{B} \sqrt{(x_i - \hat{x}_i)^2} \tag{6}$$

3. Results

In the first part of this section, the study conducted to find the orders of decomposition that optimize the MSE per sample is shown.

3.1. Optimal tensor decompositions

This exploration is performed by completing bursts of known length that have been randomly deleted from the reference database. A test of 1,000 simulations is done with 100 and 200 lost samples and using a 3 and a 7 weeks tensors. The results are given in terms of the MSE per sample, according to the exposed methodology.

The first test is done using only one decomposition and checking a set of different cores. This way the process stats are used to select the optimal core for the first decomposition, in terms of the MSE per sample. Then, to find the value of the second optimal decomposition core, the test is done with the *double decomposition* algorithm. This time, the same sets of cores is checked on the second decomposition, using the optimal configuration found on the first test for the first decomposition. This exploration already allows us to see that the method is robust for small variants in the core used on the decomposition.

3.1.1. CP case

Here the results for the optimal CP decompositions configuration are presented when the length of the bursts are 100 and 200, and for *tensorizations* of 3 and 7 weeks of data. Four cases result from the combination of burst lengths and tensor sizes. Fig.7 shows $B = 100$ with a *tensorization* of $(288 \times 7 \times 3)$; Fig.8, $B = 100$ with $(288 \times 7 \times 7)$; Fig.9 , $B = 200$ with $(288 \times 7 \times 3)$; and Fig.10 $B = 200$ with $(288 \times 7 \times 7)$. All these figures follow the same format. Figure a) shows, for different CP decomposition cores, the MSE per sample of the methodology with only one decomposition, applying the *data smoothing*, the *ramp method* and the *burst centered tensorization*. Figure b) shows the MSE per sample obtained with the *double decomposition* methodology, for different CP decomposition cores applied in the second stage, and using, as first decomposition core, the one which has given the lowest MSE value in the previous analysis. In both figures a) and b), the configuration that produces the minimum value is highlighted in red and the configurations that generate values very close to the minimum are highlighted in green .

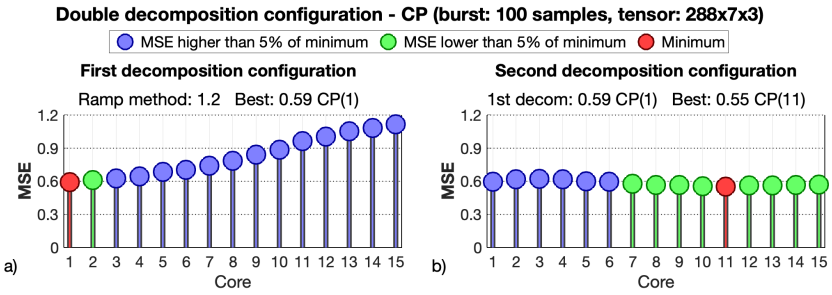

Figure 7. Test of the *double decomposition* procedure configuration with the CP model for a 100 samples burst and the 3 weeks tensor $\chi^{288\times7\times3}$. (a) shows the MSE obtained applying only the first decomposition procedure, for different core configurations. (b) shows the MSE of the *double decomposition method* for different core configurations on the second decomposition, and using the best core configuration obtained on (a) for the first decomposition, $G^{1\times1\times1}$. For each case the configuration with minimum MSE is marked in red and the configurations whose MSE rise with respect to the minimum is less than 5% are marked in green.

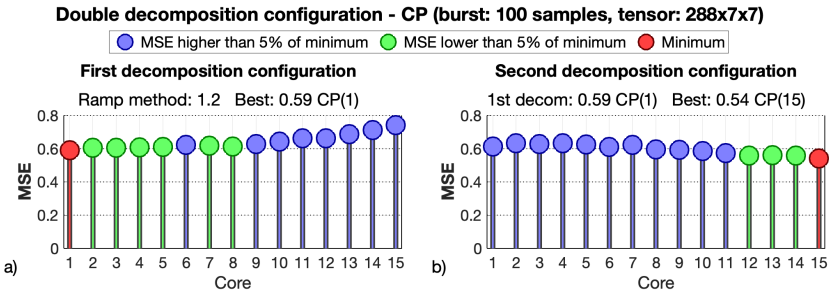

Figure 8. Test of the *double decomposition* procedure configuration with the CP model for a 100 samples burst and the 7 weeks tensor $\chi^{288\times7\times7}$. (a) shows the MSE obtained applying only the first decomposition procedure, for different core configurations. (b) shows the MSE of the *double decomposition method* for different core configurations on the second decomposition, and using the best core configuration obtained on (a) for the first decomposition, $G^{1\times1\times1}$. For each case the configuration with minimum MSE is marked in red and the configurations whose MSE rise with respect to the minimum is less than 5% are marked in green.

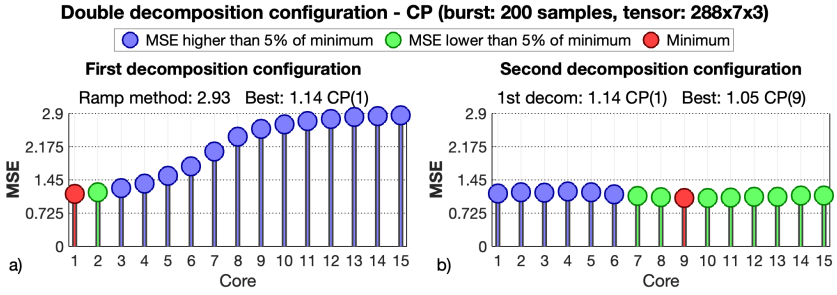

Figure 9. Test of the *double decomposition* procedure configuration with the the CP model for a 200 samples burst and the 3 weeks tensor $\chi^{288\times7\times3}$. (a) shows the MSE obtained applying only the first decomposition procedure, for different core configurations. (b) shows the MSE of the *double decomposition method* for different core configurations on the second decomposition, and using the best core configuration obtained on (a) for the first decomposition, $G^{1\times1\times1}$. For each case the configuration with minimum MSE is marked in red and the configurations whose MSE rise with respect to the minimum is less than 5% are marked in green.

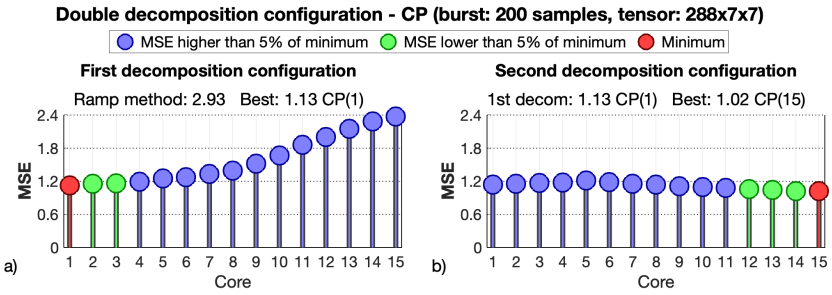

Figure 10. Test of the *double decomposition* procedure configuration with the CP model for a 200 samples burst and the 7 weeks tensor $\chi^{288\times7\times7}$. (a) shows the MSE obtained applying only the first decomposition procedure, for different core configurations. (b) shows the MSE of the *double decomposition method* for different core configurations on the second decomposition, and using the best core configuration obtained on (a) for the first decomposition, $G^{1\times1\times1}$. For each case the configuration with minimum MSE is marked in red and the configurations whose MSE rise with respect to the minimum is less than 5% are marked in green.

As important aspects to emphasize, notice that in all the tested conditions of burst and tensor sizes, the best decomposition core for the first stage is the lowest one, CP(1). Then, using the CP(1)

configuration on the first decomposition, in graph b), which shows the results for different core configurations on the second decomposition, it can be seen how the best option is to select the highest core configuration because, although in some cases it is not exactly the best, the MSE seems to have been stabilized. Therefore, for the CP method, the choice of the first decomposition core is very robust, and it has to be the lowest one. Then, for the second stage a higher decomposition core has to be selected , taking into account that there is a wide margin of acceptable configurations (values highlighted in green), because when the minimum is reached, the choice of an even higher decomposition core gives a very similar MSE.

3.1.2. Tucker case

Determining the size of the two decompositions that minimize the MSE per sample, when using Tucker decomposition, is computationally expensive and difficult to visualize. This is because there are more parameters than in the CP model to configure the decomposition core. The number of parameters depends on the tensor structure, and in the proposed 3-dimensions *tensorization* this implies having three parameters.

The experiments carried out include the cases of the burst length 100 and 200 and the *tensorizations* $\chi^{288\times7\times3}$ and $\chi^{288\times7\times7}$. Figures 7 and 8 deal with bursts of 100 missing samples and *tensorizations* of $\chi^{288\times7\times3}$ and $\chi^{288\times7\times7}$ respectively. Figures 9 and 10 deal with bursts of 200 missing samples and *tensorizations* of $\chi^{288\times7\times3}$ and $\chi^{288\times7\times7}$.

In this case, to see the optimization of Tuker model configuration graphically, one of the three parameters which have to be configured has been set, the one relative to the weeks number of data used. Then, using a matrix view, the MSE values of the combination of the other two can be further analyzed. For each figure, the first column of graphs represents the results corresponding to the first decomposition. The graphs of the second column show the MSE corresponding to the second decomposition after selecting the combination that gives the lowest value for the first one.

In all cases, the red dot represents the configuration with the lowest MSE value and the green points are those configurations with values very close to the minimum.

It is noted that the red dots are within the clusters of green points which represent quasi-optimal solutions. This means that there is a whole set of different solutions that behave in a very similar way to the optimal set. In general the best option is to select the minimum possible value on the parameter related to the number of weeks for the first decomposition and the maximum for the second one. The other two parameters seems to have more variability, but in general the parameter relative to the week day must be high, near the maximum, and the parameter relative to the day hour must be a little lower than it.

Some improvements for the method have been proposed, with the aim of refining it and obtaining better results. To see the effect of each of them the same 1,000 simulations are done without applying the improvements, followed by applying only one of them, applying some of them and finally applying them all together in Fig.15. The best results seem to be achieved with the rearrangement of the tensor using the *burst centered tensorization*, which is the best improvement if it is only applied to one of them on the methodology. Applying only the smooth process provides a little improvement on any case, not very high but constant for all the tensor and burst sizes checked.

A curious result of the *double decomposition* is that it seems to have, proportionally, a more positive effect when it is used in combination with the other options. This can be seen by comparing the MSE reduction obtained by applying only the *double decomposition* compared to using a combination of smoothing and *burst centered tensorization* or using all the improvements, specially with the Tucker model results.

With any size of tensor and burst the effect of each option is complementary to the others, which means that applying all of the improvements together provides a considerable positive impact in comparison to not using any of them in all the cases. Note that using different tensor sizes or to restoring bursts of different lengths results in a different optimal configuration of the decomposition

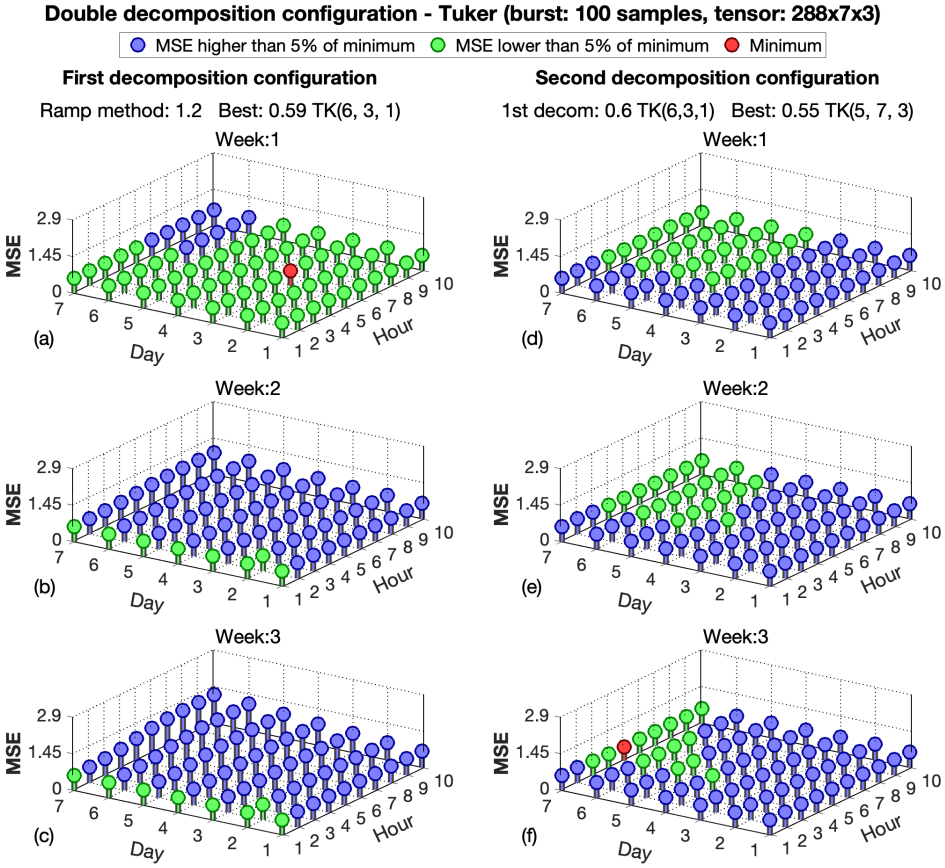

Figure 11. Test of the *double decomposition* procedure configuration with the Tucker model for a 100 samples burst and the 3 weeks tensor $\chi^{288\times7\times3}$. (a) shows the MSE obtained applying only the first decomposition procedure, for different core configurations. (b) shows the MSE of the *double decomposition method* for different core configurations on the second decomposition, and using the best core configuration obtained on (a) for the first decomposition, $G^{6\times3\times1}$. For each case the configuration with minimum MSE is marked in red and the configurations whose MSE rise with respect to the minimum is less than 5% are marked in green.

core, although with similar characteristics, Fig. 7 - Fig. 14. To show the robustness of the modified method which incorporates the *double decomposition*, the MSE obtained using different configurations for the first decomposition presented, specifically the ones found in the first column of Fig. 7 - Fig. 14, which are the optimal ones for each burst and tensor sizes, CP(1), TK(6,3,1), TK(8,5,1), TK(1,5,1) and TK(4,6,1). Table 1 also shows the consistency of algorithm results with changes in the size of the decompositions, even when combining the CP and Tucker models as well.

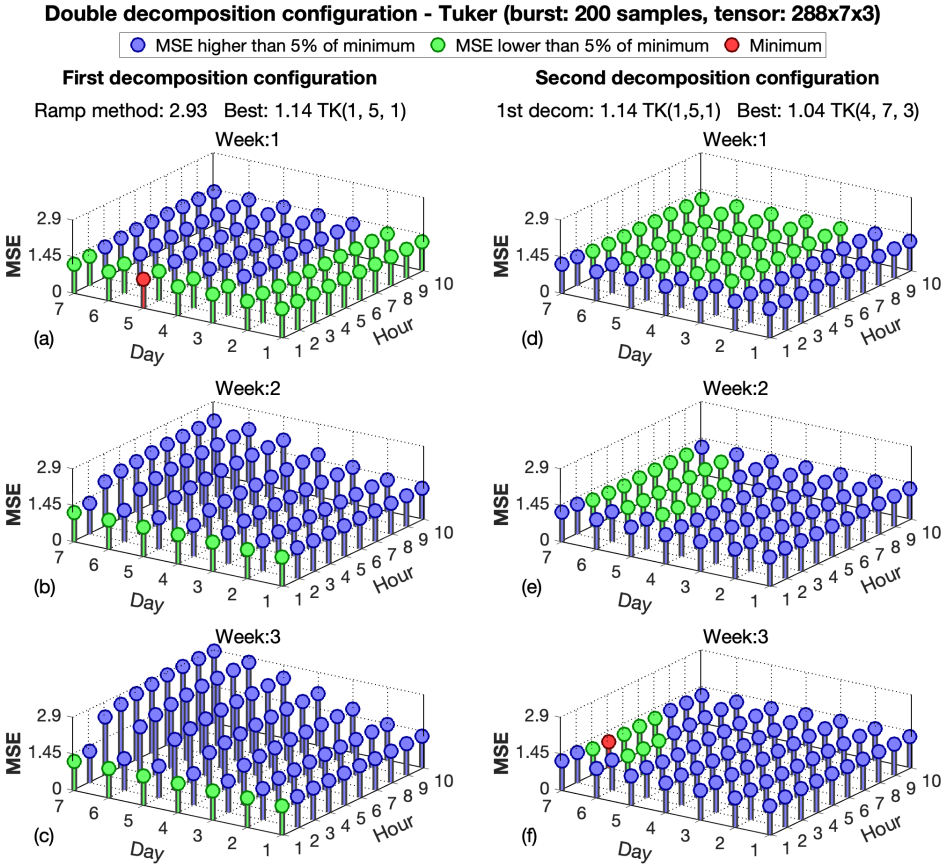

Figure 12. Test of the double decomposition procedure configuration with the Tucker model for a 200 samples burst and the 3 weeks tensor $\chi^{288\times7\times3}$. (a) shows the MSE obtained applying only the first decomposition procedure, for different core configurations. (b) shows the MSE of the *double decomposition method* for different core configurations on the second decomposition, and using the best core configuration obtained on (a) for the first decomposition, $G^{1\times5\times1}$. For each case the configuration with minimum MSE is marked in red and the configurations whose MSE rise with respect to the minimum is less than 5% are marked in green.

Table 1. MSE of the different tested methods. The results of 100 and 200 lost samples, B, are shown working with a 3 and 7 weeks of tensor size, n_w. "The best configurations in [14]" show the minimum MSE obtained for the algorithm presented in [14], with the CP and the Tucker models. "The best configurations for the proposed algorithm" show the minimum MSE obtained for different pairs of decompositions. In these cases, only the core of the first decomposition is fixed, and it is shown the minimum MSE obtained with the best core for the second decomposition.

MSE per sample	$B = 100$ $n_w = 3$	$B = 100$ $n_w = 7$	$B = 200$ $n_w = 3$	$B = 200$ $n_w = 7$
The best configurations in [14]				
optimal CP	0.87	0.80	1.70	1.58
optimal TK	0.77	0.71	1.43	1.28
The best configurations for the proposed algorithm				
1st decom: CP(1), 2nd decom: optimal CP	0.55	0.54	1.05	1.03
1st decom: TK(6,3,1), 2nd decom: optimal CP	0.57	0.52	1.14	1.02
1st decom: TK(8,5,1), 2nd decom: optimal CP	0.57	0.53	1.14	1.03
1st decom: TK(1,5,1), 2nd decom: optimal CP	0.55	0.53	1.06	1.02
1st decom: TK(4,6,1), 2nd decom: optimal CP	0.56	0.53	1.06	1.02
1st decom: CP(1), 2nd decom: optimal TK	0.54	0.52	1.04	1.00
1st decom: TK(6,3,1), 2nd decom: optimal TK	0.55	0.51	1.11	0.98
1st decom: TK(8,5,1), 2nd decom: optimal TK	0.54	0.50	1.11	0.97
1st decom: TK(1,5,1), 2nd decom: optimal TK	0.53	0.52	1.04	1.00
1st decom: TK(4,6,1), 2nd decom: optimal TK	0.55	0.50	1.11	0.97

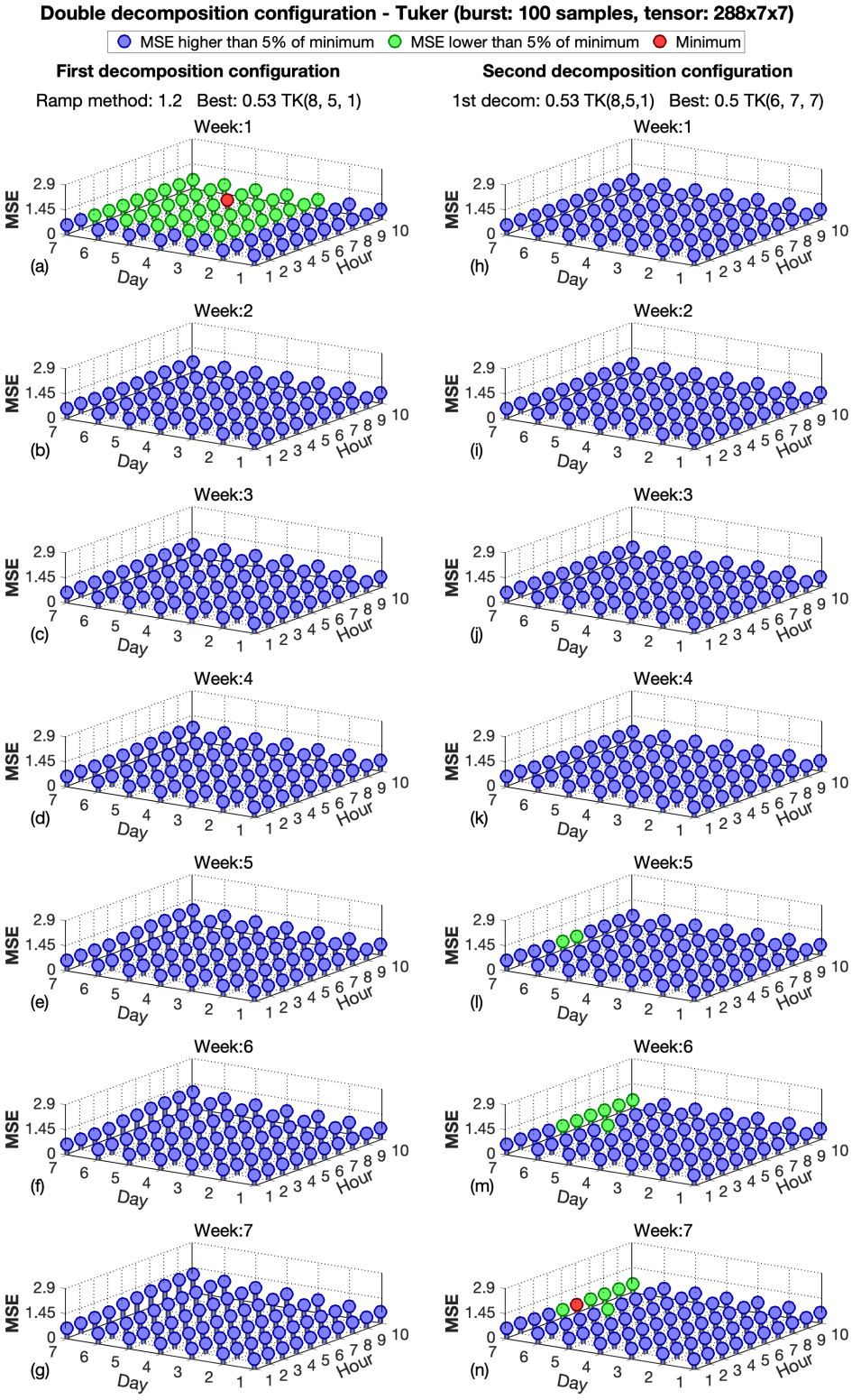

Figure 13. Test of the *double decomposition* procedure configuration with the Tucker model for a 100 samples burst and the 7 weeks tensor $\chi^{288\times7\times7}$. (a) shows the MSE obtained applying only the first decomposition procedure, for different core configurations. (b) shows the MSE of the *double decomposition method* for different core configurations on the second decomposition, and using the best core configuration obtained on (a) for the first decomposition, $G^{8\times5\times1}$. For each case the configuration with minimum MSE is marked in red and the configurations whose MSE rise with respect to the minimum is less than 5% are marked in green.

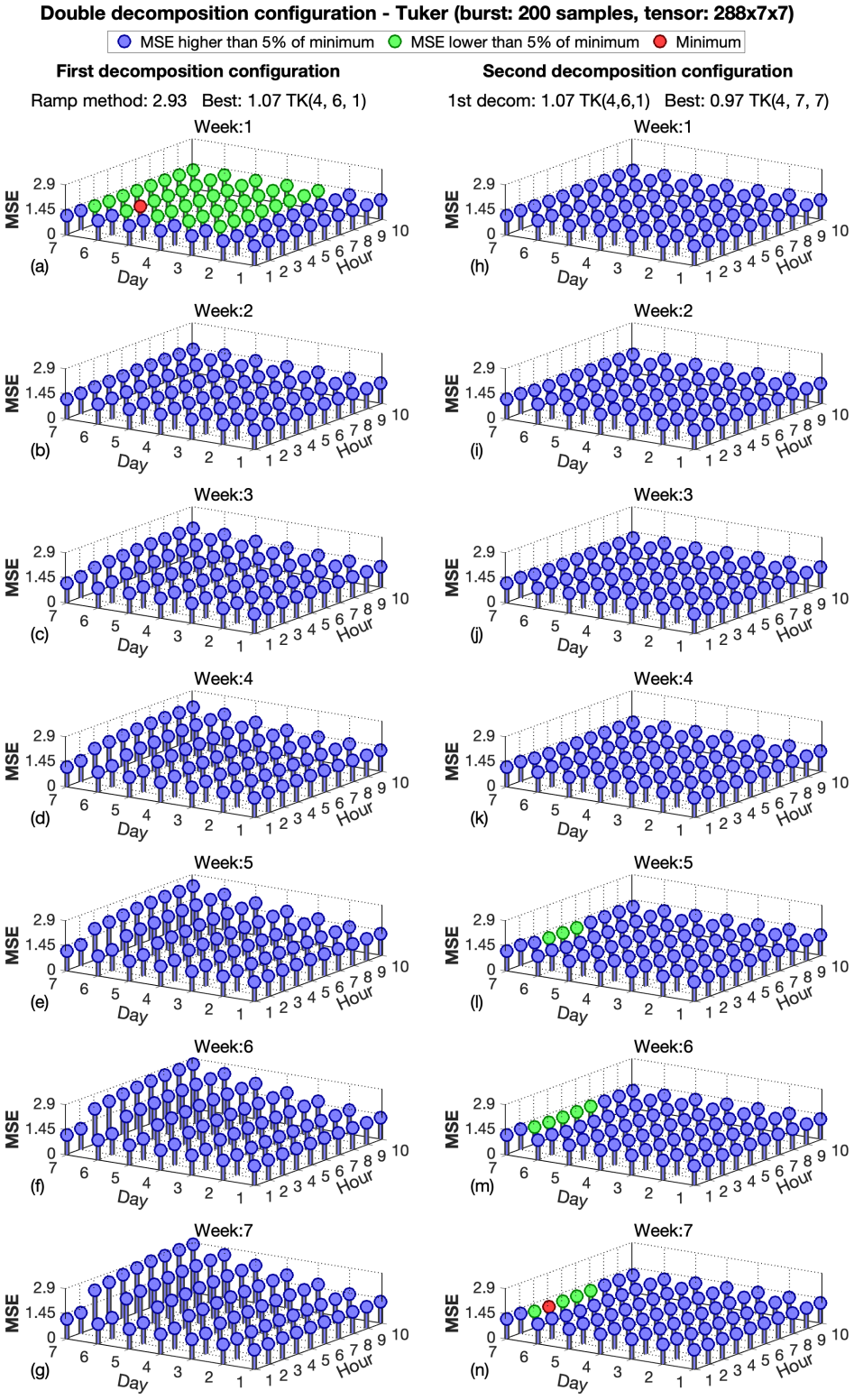

Figure 14. Test of the *double decomposition* procedure configuration with the Tucker model for a 200 samples burst and the 7 weeks tensor $\chi^{288\times7\times7}$. (a) shows the MSE obtained applying only the first decomposition procedure, for different core configurations. (b) shows the MSE of the *double decomposition method* for different core configurations on the second decomposition, and using the best core configuration obtained on (a) for the first decomposition, $G^{4\times6\times1}$. For each case the configuration with minimum MSE is marked in red and the configurations whose MSE rise with respect to the minimum is less than 5% are marked in green.

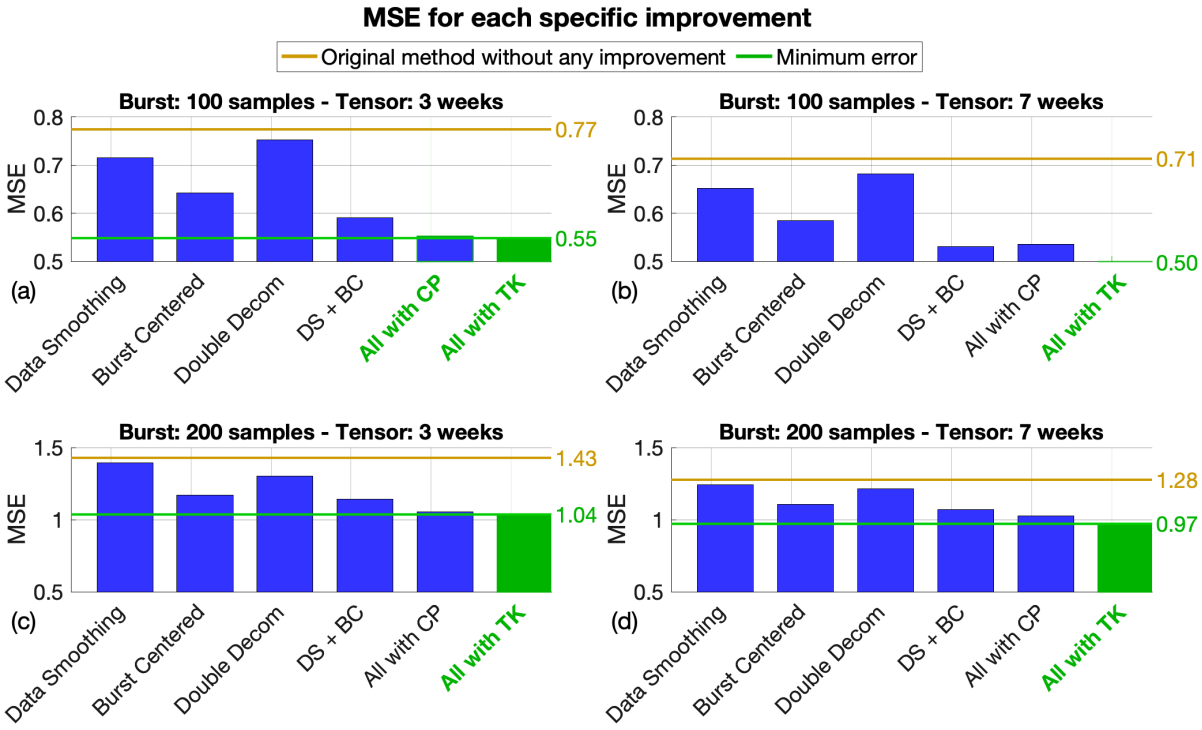

Figure 15. MSE of the proposed improvements. All of them are tested with 100 and 200 samples of data burst lost and with 3 and 7 weeks of *tensorized* data. The orange line indicates the best result obtained in [14]. The *Smooth* is the result of applying only the smooth process to the signal before *tensorizing* it. The *burst centered tensorization* is the result of the rearrangement of the tensor according the burst location. The *Double decom* is the result of applying the decomposition two times, with $G^{4\times6\times1}$ and $G^{4\times7\times7}$ for Tuker, and $G^{1\times1\times1}$ and $G^{15\times15\times15}$ for CP. The *DS-Bc* is the result of combining the data smoothing and the *burst centered tensorization* without using the double decomposition. The *All with CP* and the *All with TK* are the results of applying all the proposed algorithm with the CP and the Tucker models respectively.

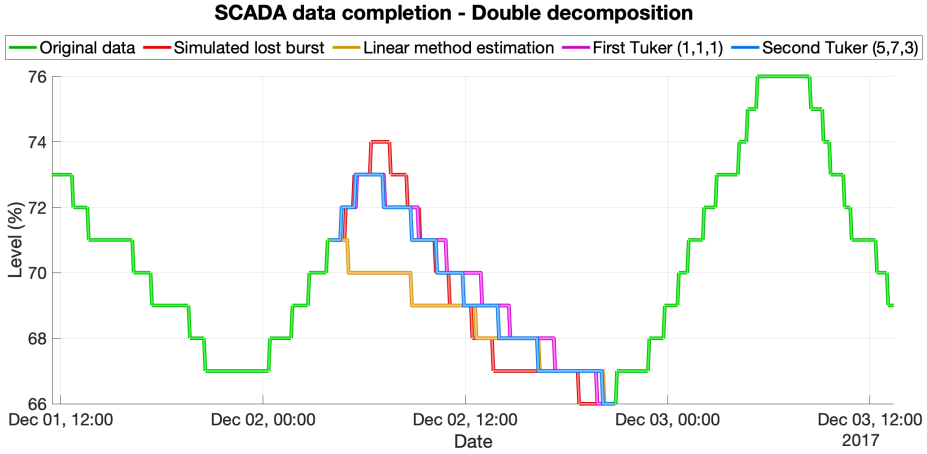

Figure 16. Example of the reconstruction methodology with double decomposition. The green line shows the original data and the red line the burst of lost samples. The orange line is the linear estimation with the *ramp method*. The purple line is the result of the first *tensorization* procedure with Tuker using $G^{4\times6\times1}$. The blue line shows the result of the second *tensorization* procedure with $G^{4\times7\times7}$.

4. Conclusions

Completing data lost in bursts remains a difficult challenge and is where most data completion methods fail. However, data being lost in bursts is quite common. It is often associated with the failure of a component involved in capturing or transmitting the data. In the contribution of [14] it is presented an *ad hoc* data completion method is presented to recover data lost in bursts that outperforms the methods available in the literature for the proposed application. This work improves the method in [14], which is taken as a reference for the new algorithm evaluation and for comparison purposes. It is also often difficult to evaluate data completion methods. In this article, intensive experiments on a verified database were carried out by erasing data statistically and comparing the results of the algorithms with the original data are carried out. The method incorporates fundamental novelties such as a new *tensorization* method, the *burst centered tensorization*, and the application of two tensor decomposition, one after the other.

Note that the MSE corresponding to a 100 samples burst falls from 0.71, the best result obtained in [14], to 0.50, the best result obtained with this new methodology. Thus, approximately a 39.5% of reduction is achieved. In the case of a 200 samples burst the MSE falls from 1.28 to 0.97 which is approximately a 24.2% of reduction.

A signal reconstruction example of this procedure is shown in Fig.16 using $G^{4\times6\times1}$ and $G^{4\times7\times7}$, the best core configurations for the double decomposition according to the tests for the Tucker model in the case of a 200 samples of burst length.

Author Contributions: conceptualization, P.M.-P.and A.M.-S.; methodology, P.M.-P., M.S.-S. and A.M.-S; software, P.M.-P. and A.M.-S. ; validation, P.M.-P., M.S.-S and A.M.-S.; formal analysis, P.M.-P., M.S.-S and A.M.-S; investigation, P.M.-P. and A.M.-S.; resources, P.M.-P. and M.S.-S; writing–original draft preparation, P.M.-P. and A.M.-S.; writing–review and editing, P.M.-P. and A.M.-S.; supervision, P.M.-P. and M.S.-S.; funding acquisition, P.M.-P. and M.S.-S.

Funding: Financial support by the Agency for Management of University and Research Grants (AGAUR) of the Catalan Government to Arnau Martí-Sarri is gratefully acknowledged.

Acknowledgments: We thank the company Aigües de Vic S.A. for giving us access to their databases to perform this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Langhammer, J.; Česák, J. Applicability of a Nu-Support Vector Regression Model for the Completion of Missing Data in Hydrological Time Series. *Water* **2016**, *8*. doi:10.3390/w8120560.
2. Ahlheim, M.; Frör, O.; Luo, J.; Pelz, S.; Jiang, T. Towards a Comprehensive Valuation of Water Management Projects When Data Availability Is Incomplete — The Use of Benefit Transfer Techniques. *Water* **2015**, *7*, 2472–2493. doi:10.3390/w7052472.
3. Zhao, Q.; Zhu, Y.; Wan, D.; Yu, Y.; Cheng, X. Research on the Data-Driven Quality Control Method of Hydrological Time Series Data. *Water* **2018**, *10*. doi:10.3390/w10121712.
4. Ekeu-wei, I.T.; Blackburn, G.A.; Pedruco, P. Infilling Missing Data in Hydrology: Solutions Using Satellite Radar Altimetry and Multiple Imputation for Data-Sparse Regions. *Water* **2018**, *10*. doi:10.3390/w10101483.
5. Blanch, J.; Puig, V.; Saludes, J.; Quevedo, J. Arima models for data consistency of flowmeters in water distribution networks. *IFAC Proceedings Volumes* **2009**, *42*, 480–485.
6. Lamrini, B.; Lakhal, E.K.; Le Lann, M.V.; Wehenkel, L. Data validation and missing data reconstruction using self-organizing map for water treatment. *Neural Computing and Applications* **2011**, *20*, 575–588.
7. Puig, V.; Ocampo-Martinez, C.; Pérez, R.; Cembrano, G.; Quevedo, J.; Escobet, T. *Real-Time Monitoring and Operational Control of Drinking-Water Systems*; Springer, 2017.
8. Acar, E.; Dunlavy, D.M.; Kolda, T.G.; Mørup, M. Scalable tensor factorizations for incomplete data. *Chemometrics and Intelligent Laboratory Systems* **2011**, *106*, 41–56.
9. Signoretto, M.; Van de Plas, R.; De Moor, B.; Suykens, J.A. Tensor versus matrix completion: a comparison with application to spectral data. *IEEE Signal Processing Letters* **2011**, *18*, 403.

10. Mørup, M. Applications of tensor (multiway array) factorizations and decompositions in data mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **2011**, *1*, 24–40.
11. Kolda, T.G.; Bader, B.W. Tensor decompositions and applications. *SIAM review* **2009**, *51*, 455–500.
12. Cichocki, A.; Mandic, D.; De Lathauwer, L.; Zhou, G.; Zhao, Q.; Caiafa, C.; Phan, H.A. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE Signal Processing Magazine* **2015**, *32*, 145–163.
13. Comon, P. Tensors: a brief introduction. *IEEE Signal Processing Magazine* **2014**, *31*, 44–53.
14. Marti-Puig, P.; Martí-Sarri, A.; Serra-Serra, M. Different Approaches to SCADA Data Completion in Water Networks. *Water* **2019**, *11*, 1023.
15. Tucker, L.R. Some mathematical notes on three-mode factor analysis. *Psychometrika* **1966**, *31*, 279–311.
16. Carroll, J.D.; Chang, J.J. Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. *Psychometrika* **1970**, *35*, 283–319.
17. De Lathauwer, L.; De Moor, B.; Vandewalle, J. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications* **2000**, *21*, 1253–1278.
18. Harshman, R.A. Foundations of the PARAFAC procedure: Models and conditions for an" explanatory" multimodal factor analysis. *Working Papers in Phonetics* **1970**, *UCLA Working Papers in Phonetics, 16*, 1–84.
19. Sørensen, M.; Lathauwer, L.D.; Comon, P.; Icart, S.; Deneire, L. Canonical polyadic decomposition with a columnwise orthonormal factor matrix. *SIAM Journal on Matrix Analysis and Applications* **2012**, *33*, 1190–1213.
20. Sidiropoulos, N.D.; De Lathauwer, L.; Fu, X.; Huang, K.; Papalexakis, E.E.; Faloutsos, C. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing* **2017**, *65*, 3551–3582.
21. Kolda, T.G. Multilinear operators for higher-order decompositions. Technical report, Sandia National Laboratories, 2006.