

POVEZAVA MERILNIH SIGNALOV IN MATLABA PREKO PROCESNEGA STREŽNIKA

S. Peršin, B. Tovornik, N. Muškinja

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Maribor, Slovenija

Ključne besede: okolja industrijska, okolja raziskovalna, procesi proizvodni, avtomatizacija, signali merilni, strežniki procesni, PLC krmilniki logični programirljivi, povezave, MATLAB oprema programska simulacijska, SCADA krmiljenje nadzorno in zajemanje podatkov, merjenje, optimiranje, komunikacije v času realnem, OLE vgraditev in povezava objektov, OPC OLE krmiljenja procesov

Izvelek: Sodobni sistemi za avtomatizacijo proizvodnih procesov uporabljajo industrijsko testirane, na tržišču dostopne ter uveljavljene proizvode kot so senzori, aktuatorji, krmilniki (PLC - Programmable Logic Controller) in SCADA (Supervisory Control and Data Acquisition) sistemi. Le-ti zagotavljajo zanesljivo in varno delovanje proizvodnega procesa. Takšni sistemi so zelo omejeni glede na možnosti raziskav, optimizacije, testiranja in simulacije, kar pa omogočajo sistemi, znani iz raziskovalnega okolja. Povezava takega industrijskega okolja s prožnim, a nestabilnim raziskovalnim okoljem je osnovni cilj predstavljenega dela. Za povezavo industrijskega sistema z Matlab/Simulink programom je bila uporabljena OPC tehnologija, ki je pogosto sestavni del sistemov industrijske avtomatizacije in je tam že dobro uveljavljena. OPC je okrajšava za OLE for Process Control, kjer je OLE oznaka za Microsoftovo tehnologijo Object Linking and Embedding. Uporaba Matlab simulacijskega programa in njegovih orodij omogoča veliko prožnost pri raziskavah, analizah in optimizaciji, kar v današnjih SCADA sistemih ni zagotovljeno. V testnem sistemu je strežnik zajemal podatke iz krmilnika, preveril kvaliteto in morebitno spremembo vrednosti podatkov ter jih posredoval odjemalcu, ki jih je po obdelavi poslal nazaj po isti poti. Za namene "worst case" testiranja se je uporabilo sinhrono branje, kjer se je vsak podatek vsakič spremenil, na istem računalniku pa so tekli en strežnik, dva odjemalca, SCADA ter Matlab program. Bistvena prednost predstavljene rešitve je integracija računskih, simulacijskih ter grafičnih zmogljivosti Matlab z robustnostjo industrijskega sistema. Rešitev je univerzalna, omogoča sprotno obdelavo podatkov in dostop do podatkov drugim programskim orodjem, ne da bi pri tem fizično posegali v industrijski sistem za avtomatizacijo.

Connection Between Measurement Signals and MATLAB Using Process Server

Key words: industrial environments, research environments, fabrication processes, automation, measurement signals, process servers, PLC, Programmable Logic Controllers, connections, MATLAB simulation software, MATrix Laboratory simulation software, SCADA, Supervisory Control And Data Acquisition, measurements, optimization, real-time communications, OLE, Object Linking and Embedding, OPC, OLE for Process Control, Object Linking and Embedding for Process Control

Abstract: The automation of processes has reached the point where one cannot go without applying sensors, actuators, PLC (Programmable Logic Controllers) and SCADA (Supervisory Control and Data Acquisition) systems that enable acquisition of large quantities of data. The process automation is particularly important in isolated rooms where physical presence of humans is undesirable, as for example in the production of electronic components or in the pharmaceutical industry. Such manufacturing systems operate with various degrees of success and are not modified very often. Every alternation of any industrial process is followed by extensive documenting, testing and validating of the new system. Quite often the production must be stopped which gives a rise to substantial costs for entire operation. On the other hand there are a number of research laboratories where various approaches are constantly a subject to change, development and testing in order to bring about improvements. The research work often implements mathematical and simulation software such as Matlab (Matrix Laboratory) which enables solving of complex mathematical operations, simulations, optimisations, etc. However, their main disadvantage is that they are substantially limited in their connectivity to other systems and applications. Manufacturing processes include many procedures where large amounts of data could be processed, examined and optimised: regulation loops, optimal production structuring, calculation of illumination in various points, etc. The equipment that is used in the industry, such as PLC and SCADA, is not suitable for performing such data processing. Instead, Matlab would be much better if only it were possible to feed it with real-time data from the manufacturing process. By applying this technology a wide spectrum of possibilities for testing and analysis emerges which can contribute to the improvement of the process. The system can also be upgraded to perform a real-time parallel optimisation procedure. In this configuration the manufacturing process would be running independently whereas Matlab would be performing in parallel all the necessary mathematical operations, the results of which would be fed back into the process. There are two difficulties in implementing such real-time connections: the problem of connectivity and the difficulty of presenting a universal solution enough that would not require altering our existent manufacturing systems, especially when new research requires updating or changing our existing configurations. One possible solution would be to use the OPC standard and the associated technology. OPC means OLE for Process Control (OLE - Object Linking and Embedding) and represents an international industrial standard (Specifications v.1.0a) that is based on the Microsoft's OLE/COM/DCOM technology. It provides a unified communication interface based on Plug&Play between various devices of process control. The OPC interface thus belongs to one of the new technologies applied to connecting and interfacing systems of process control which is based on the component technology. The main properties of the component technology can be summarised by the properties of OLE automation which enable that one application can contain objects from other applications, the linking of both applications results in data being simultaneously updated in both, and non-native objects can be executed or changed inside the application where they are embedded. In the usual systems of process control each SCADA software has to have its own piece of software code in order to be able to communicate with the kernel of the I/O driver or with any other part of the software code that is used to communicate with the I/O unit. In other words: each SCADA system needs drivers in order to be able to communicate with other devices and each device connected to the system must have its own driver. If a system is connected to another SCADA system then this system needs its own drivers again for all the devices that will communicate with it and that are connected to it. The number of required drivers can be calculated by multiplying the number of I/O units with the number of SCADA packets. The result is a very high number and thus an inelegant solution. As a solution, the OPC creates a 'software bus' where applications only need to be aware on how to access data on the OPC. Testing system consisted of an OPC server which captured data from the

controller, it checked the data's quality, evaluated it for a change of state, and then forwarded it to the OPC client who returned it along the same route once it was processed. For the 'worst case' scenario synchronous reading was used where each piece of data was changed every time and the same computer ran a server, two clients, as well as the SCADA and Matlab software. The measurements performed show that due to the fixed dead time the worst results are obtained when transferring a single piece of data only. The results show a rate of approximately 1000 pieces of data per second can be expected with large quantities of data. The process data is available to Matlab virtually at the same time as to the SCADA system which makes the application suitable for industrial environments. The proposed solution is especially suitable for manufacturing processes with slower rates of change since here it can also be used for process control. Main advantages of proposed solution are the integration of the numerical, computational and graphical powers of MATLAB® with the robustness of industrial system, universality, on-line data processing and accessibility to other software programs, without any physical connection or disturbance of the manufacture automation system.

1. Uvod

Proizvodnja je dandanes avtomatizirana do te mere, da si je več ne znamo predstavljati brez obilice senzorjev, aktuatorjev, krmilnikov (PLC - Programmable Logic Controller) ter SCADA (Supervisory Control and Data Acquisition) sistemov /12/, ki omogočajo veliko število informacij. Prav posebej visok nivo avtomatizacije je v procesih, ki potekajo v t.i. čistih prostorih, v katerih je fizična prisotnost ljudi nezaželjena, kot je to primer pri proizvodnji elektronskih komponent /21/ ali zdravil. Ti sistemi v proizvodnih procesih bolj ali manj uspešno delujejo in se predvsem ne spreminjajo prepogosto /19/. Vsak poseg v proizvodni proces je namreč povezan vsaj z obširnimi dokumentiranjem, testiranjem ter validiranjem, tudi s strani certifikacijskih organizacij, kot sta npr. ISO (International Standard Organization) ter FDA (Food and Drug Administration), če ne tudi z zaustavitvijo proizvodnje, kar ima za posledico velike skupne stroške posega /20/.

Na drugi strani pa je svet raziskovalnih laboratorijev, ki je podvržen nenehnim spremembam, razvoju ter testiranjem različnih pristopov v smeri izboljšav. V raziskovalno okolje so velikokrat vključeni tudi matematični in simulacijski programi kot je Matlab (Matrix Laboratory), ki omogoča reševanje zahtevnih matematičnih operacij, simulacije, optimizacije itd., vendar imajo omejene možnosti povezovanja /18/.

V proizvodnem procesu je veliko postopkov s podatki, ki so zanimivi za obdelavo, raziskave ter morebitno optimiranje: od regulacijskih zank, optimalnega razvrščanja proizvodnje pa do npr. izračunavanja osvetlitve v posameznih točkah. Zaradi velike matematične zahtevnosti omenjenega že instalirana oprema (krmilniki in SCADA) za tovrstne operacije ni primerna, primeren pa je Matlab /10/, v katerega je potrebno sprotno prenašati podatke iz proizvodnega procesa. Z uporabo takšne rešitve se kaže širok spekter možnosti testiranja ter analiz v smeri izboljšanja delovanja procesa, kar se lahko nadgradi tudi s sprotnim paralelnim optimizacijskim postopkom, kjer industrijski proces teče samostojno, hkrati pa se paralelno v Matlabu izvajajo potrebne matematične operacije, katerih rezultati se sporočajo nazaj v proces. Pri implementiranju On-line povezave je zraven problema poveztljivosti prisotna tudi problematika univerzalnosti rešitve ter predvsem nespreminjanja obstoječega industrijskega sistema tudi v primeru morebitnih dopolnitev ali sprememb na raziskovalnem nivoju.

Možna rešitev je uporaba OPC standarda /22/ in tehnologije, ki je s tem povezana. OPC je okrajšava za OLE for Process Control, kjer je OLE oznaka za Microsoftovo tehnologijo Object Linking and Embedding. V članku je predstavljena rešitev, v kateri se znotraj okolja Matlab uporabi OPC odjemalec, ki izmenjuje podatke z že obstoječim OPC strežnikom na nivoju povezave PLC-SCADA, kar je prikazano na sliki 1. OPC se zmeraj bolj uporablja v sistemih industrijske avtomatizacije /12/ /14/ /1//3//2/, prav razširjenost uporabe pa daje rešitvi potrebno univerzalnost.



Slika 1: Uporaba OPC za povezavo industrijskega in raziskovalnega okolja

V naslednjih poglavjih bo predstavljeno nekaj pogostejše uporabljenih tehnik za On-line matematično analizo procesnih podatkov, povzete glavne lastnosti OPC in predstavljena zasnova in izvedba povezave Matlaba z industrijskim krmilnikom, skupaj z rezultati eksperimenta. V zaključku je podana analiza z povzetkom glavnih lastnosti takega pristopa, podane pa so tudi smernice za nadaljevanje dela.

2. Nekaj do sedaj uporabljenih rešitev za On-line matematično analizo procesnih podatkov

Uporaba možnosti vključevanja lastnih programov v enem višjih programskih jezikov, kar ponuja večina SCADA programov, je na prvi pogled enostavna rešitev za dodatno matematično obdelavo. Težave se lahko pojavijo pri programiranju zahtevnih matematičnih operacij.

Ena pogostejše uporabljenih rešitev za zajem realnih podatkov v Matlab je uporaba posebnih PC I/O modulov /4/, /9/, ki jih podpira Matlab (npr. Burr-Brown ali National Instruments). Prednost takšne povezave je zraven produktne usklajenosti predvsem hitrost operiranja z podatki, ki je lahko manjša od 1 ms/podatek. Vendar pa gre za tipično laboratorijsko opremo, ki se jo v proizvodnih procesih sreča

le redko, kar pomeni zraven nakupa strojne opreme še težave pri implementiranju le-te v sam proizvodni proces. Takšno kartico je potrebno fizično povezati na obstoječo senzoriko in sicer vzporedno ali zaporedno s krmilniško opremo, kar pomeni že omenjene težave v povezavi s stroški zaustavitve proizvodnje, dokumentiranjem ter validiranjem. Poudariti je potrebno, da je težav z izvedbo takšnega posega ni, ko so cilji ter rezultati posodobitve natančno definirani, do česar pa je zopet težko priti brez On-line testiranja ter prezentiranja rezultatov na dotičnem procesu. Podobna je rešitev z uporabo merilnih instrumentov z možnostjo GPIB ali VISA komunikacije, ki jo podpira tudi Matlab.

V Matlabu je omogočen dostop do serijskega vodila, kar ponuja možnost izdelave lastnega API (Application Programming Interface), ki »prisluškuje« prometu po mreži krmilnikov in po potrebi tudi pošilja podatke. Prednost takšne ideje je, da ne poseže direktno v proces. Pomanjkljivost je časovno zahtevna izdelava vmesnika za industrijsko vodilo ter neuniverzalnost rešitve /10/.

Za obravnavano problematiko so se rešitve zmeraj bolj nagibale proti programskim rešitvam, ki ponujajo bolj izmenjavo podatkov kot pa dodatno zajemanje podatkov, saj se meritve tako ali tako kontinuirano opravljajo za potrebe krmilnikov in SCADA in zato ni potrebno podvajanje. Najenostavnejša izmenjava podatkov je mogoča preko ASCII datoteke, v katero vpisujeta oba, SCADA in Matlab. Rešitev je v omejenem obsegu univerzalna, vendar so potrebni posegi v SCADA program, pa tudi prenos ni hiter.

S pomočjo OLE (Object Linking and Embedding) je mogoča povezava Matlaba z MS Excelom, kar omogoča standardni Matlabov Toolbox Excel Link. Excel podpira DDE (Dynamic Data Exchange), ki ga podpirajo tudi SCADA sistemi, kar ponuja možnost povezave, ki je med bolj univerzalnimi, vendar se DDE na področju procesnega vodenja nikoli ni povsem uveljavil zaradi neučinkovitosti in nezanesljivosti pri prenašanju velikih količin podatkov /5/. Dodatna težava omenjenega pristopa je potreben poseg v SCADA sistem, iz katerega je potrebno pošiljati DDE podatke (ob predpostavki, da SCADA ne bazira na DDE). Tak poseg pomeni spremembo funkcionalnosti in s tem potrebo po validacijskem postopku.

Uporaba OPC (OLE for Process Control) za izmenjavo podatkov med Matlabom in industrijskim procesom predstavlja logično nadaljevanje predstavljenih tehnik in je opisana v nadaljevanju.

3. OPC

3.1. Uvod

Za kratico OPC se skriva ime OLE for Process Control (OLE - Object Linking and Embedding) in predstavlja mednarodni industrijski standard (specifikacija v1.0a), ki temelji na Microsoftovi OLE/COM/DCOM tehnologiji in zagotavlja

skupen komunikacijski plug&play vmesnik med različnimi napravami na področju procesnega vodenja. OPC vmesnik je ena izmed novih tehnologij pri povezovanju in komunikaciji v sistemih procesnega vodenja, ki temelji na komponentni tehnologiji. Osnovna načela komponentne tehnologije se lahko opišejo s principi OLE avtomatizacije, ki določajo možnost, da ena aplikacija vsebuje objekt iz druge aplikacije, da preko povezovanja prihaja do osveževanja podatkov med njima ter da je možno objekte spreminjati in poganjati znotraj aplikacije, ki jih vsebuje.

3.2. Problem povezljivosti v sistemih procesnega vodenja

V običajnih sistemih procesnega vodenja potrebuje vsaka SCADA programska oprema lasten kos programske kode za komunikacijo z jedrom I/O gonilnika oz. tistim delom programske kode, ki zna komunicirati z I/O enoto /14/. Če se poenostavi: vsak SCADA sistem potrebuje za komunikacijo z napravami gonilnike, ki jih je toliko, kot je naprav priključenih na sistem. Če se sistem povezuje z drugim SCADA sistemom, potrebuje ta sistem zopet svoje gonilnike za vse naprave s katerimi bo komuniciral in ki so priključene na ta sistem. Izračuna se lahko, koliko različnih gonilnikov je potrebno, tako da se pomnoži število I/O enot s številom SCADA paketov; rezultat pa je zelo visoka številka ter nepraktična rešitev. Tovrste povezave SCADA sistemov v celoto se zato izvedejo večinoma le znotraj istih proizvajalcev procesne in programske opreme (Siemens, GE, Omron...).

OPC vmesnik se v kos programske kode za komunikacijo z jedrom I/O gonilnikov ne pogloblja, saj se z njimi v večini primerov ukvarjajo protokoli na nižjem nivoju kot so Field-Bus, DeviceNet, Profibus, LonWorks /11/. Problem je, da vsak gonilnik uporablja svoj vmesnik za komunikacijo s periferno napravo oziroma aplikacijo, kar onemogoči več aplikacijam hkrati komunicirati z isto periferno napravo. OPC pa v nasprotju z opisanim ustvari t.i. programsko vodilo (»software bus«), kjer morajo aplikacije, oz. programerji aplikacij, vedeti samo kako dostopati do podatkov na OPC programskem vodilu, ki jih posredujejo periferne naprave. Aplikacije so tako preprostejše, manjše in enostavnejše za uporabo. Gonilniki perifernih naprav (OPC strežniki) morajo tako samo znati prenašati informacije iz perifernih naprav na OPC vodilo v določenem formatu.

Vendar univerzalnost prenosa podatkov ni edina prednost, ki jo ponuja OPC vmesnik. Prvotno namenjen reševanju problematike I/O gonilnikov, se je OPC vmesnik izkazal dovolj fleksibilen, da se poleg povezovanja perifernih naprav s SCADA sistemi uporablja tudi pri povezovanju z aplikacijami »višje navzgor« vse do sistemov poslovnega odločanja.

3.3. Tehnologija OPC vmesnika

Povezovanje z OPC vmesnikom se vrši s pomočjo programskih vmesnikov, ki temeljijo na COM (Component Object

Model) tehnologiji, različna vodila, kot sta Fieldbus in Profibus, pa se ukvarjajo s fizično in mrežno povezavo. COM je Microsoftova osnova za pisanje API-jev (Application Programming Interface). Ob upoštevanju, da je zelo široko uporabljena, omogoča tudi zelo dobro navpično povezovanje, kjer je potrebno povezati OPC strežnik ter OPC odjemalec, ki je običajno že integriran v samem SCADA sistemu. To vključuje tudi DCOM tehnologija (Distribuirani COM), ki omogoča ustvarjanje aplikacij, ki temeljijo na mrežnem povezovanju. Še najbližje takšni strukturi prenosa podatkov se je približal DDE vmesnik, vendar se na področju procesnega vodenja nikoli ni povsem uveljavil zaradi neučinkovitosti in nezanesljivosti pri prenašanju velikih količin podatkov. Mnogi proizvajalci programske opreme so raje uporabljali lastne API-je, kot pa DDE vmesnik za prenos procesnih podatkov iz periferne enote v programsko opremo ali iz ene v drugo programsko opremo.

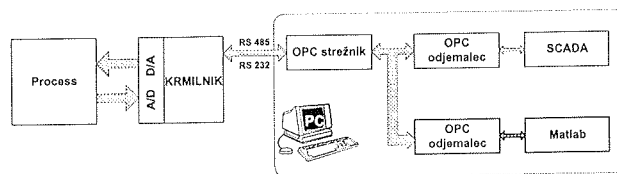
OPC je dovolj fleksibilen za delovanje v različnih plasteh sistemov, kot tudi dovolj dober za uporabo preko mreže. Učinkovitost OPC vmesnika je zagotovljena, saj lahko le-ta prenaša na stotine in tisoče podatkov z eno transakcijo, kar DDE ne omogoča. Vsaka transakcija lahko vključuje še mnogo različnih spremenljivk. Tako je vsaka transakcija OPC vmesnika sestavljena iz podatka, časovne znamke podatka ter informacije o kvaliteti podatka. Z DDE vmesnikom bi se za to potrebovale tri transakcije. OPC strežniki so lahko integrirani v samem procesu, lahko so lokalni (Local) ali pa so daljinski (Remote). Glede na učinkovitost prenosa podatkov je zelo pomembno kje se nahaja OPC strežnik, saj omogoča v prvem primeru (v procesu) milijon transakcij na sekundo, v drugem (Local) okoli tisoč in v tretjem (Remote) okoli sto.

Ponujena je tudi možnost, da lahko proizvajalci oz. programerji ustvarijo in/ali nadgradijo enostaven in učinkovit gonilnik, ki podpira delovanje s perifernimi napravami ne glede na proizvajalca. S tem je omogočeno, da OPC strežnik univerzalno prenaša real-time podatke procesnih spremenljivk kot so temperatura, tlak, pretok, pozicija, hitrost, itd. Zajem podatkov je možno še poenostaviti, saj OPC aplikacija dovoljuje definiranje ene ali več »skupin« (Groups), v katerih se potem nadzirajo posamezne želene spremenljivke. OPC odjemalec, ki je odjemalec podatkov iz OPC vodila in posrednik podatkov v SCADA sistem lahko potem izvaja sinhrono ali asinhrono branje podatkov želenih spremenljivk v skupini.

Poudarjene funkcije pri povezovanju in komunikaciji z OPC vmesnikom so zraven on-line dostopa do podatkov (On-line Data Access) še upravljanje z alarmi in dogodki (Alarm and Event Handling), beleženje podatkov (Historical Data Access), zaščita dostopa (Security) in prenos podatkov pri šaržnih procesih (Batch). Zadnji dve specifikaciji sta še v fazi dogovorov, razvoja in standardizacije, v fazi zaključnih dogovorov pa so specifikacije za področje OPC XML ter nova specifikacija OPC Data Access 3.0 /23/.

4. Izveden eksperiment

Z matematičnim orodjem Matlab smo v okolju Simulink zgradili testno okolje za eksperiment, katerega cilj je bil realizirati On-line prenos procesnih podatkov v Matlab iz klasičnega industrijskega krmilnika (slika 2). Zajem podatkov v Matlab je sicer mogoč s pomočjo posebnih I/O računalniških kartic, kar pa je rešitev, ki večinoma v realnih industrijskih sistemih ni mogoča. Razlogi tičijo v že instalirani opremi, (ne)robustnosti, internih standardih, itd. V tovarniškem okolju je uporaba industrijskega krmilnika in SCADA sistema največkrat edina možnost. Po drugi strani pa je programski paket Matlab s Simulinkom zelo močno matematično orodje, ki omogoča boljšo matematično obdelavo podatkov kot katerikoli SCADA sistem. Zato je smiselno, da se ga vključi v sisteme procesnega vodenja.



Slika 2: Testno okolje za eksperiment

Kot tipičen predstavnik industrijskih krmilnikov je bil porabljen krmilnik S7-414-2DP proizvajalca Siemens. Namen predstavljene rešitve je branje in vpisovanje on-line procesnih podatkov iz krmilnika v Matlab. Za programiranje krmilnika je bil uporabljen programski paket Simatic STEP7, povezava z računalnikom pa je potekala preko vmesnika MPI/RS 232 (Multi Point Interface), kjer je bila hitrost omejena na 19 200 bps. Še boljše rezultate pri prenosu podatkov bi se lahko doseglo z uporabo Profibus vodila, kjer je hitrost prenosa bistveno višja.

Za programsko zajemanje podatkov je bil uporabljen program KEPServerEX OPC strežnik neodvisnega proizvajalca KepWare. Potrebno je bilo ustvariti kanal po katerem poteka komunikacija, izbrati napravo s katere se zajema podatke ter določiti skupino spremenljivk, ki se jih naj nadzoruje. S programom OPC Quick Client istega proizvajalca je možno preklopiti OPC strežnik v monitoring način in tako vnaprej preveriti, če OPC povezava oziroma komunikacija deluje.

Za OPC odjemalec je bil uporabljen OPC Client for Matlab, ki omogoča komunikacijo med OPC in Matlab-om. Gre za nekaj dodatnih ukazov, ki so na razpolago v okolju Matlab in omogočajo povezavo na katerikoli OPC strežnik. Za konfiguracijo tega programa je potrebno vpisati določene ukaze v komandno okno programa Matlab, kot je prikazano na sliki 3 za primer komunikacije z enim vhodom in enim izhodom.

```
i=0;
while (i==0);
hr=mxOPC('open','KEPware.KEPServerEx.V4','localhost',10);
mode=mxOPC('readmode','cache');
hr=mxOPC('setdoublecache','Channell.Device1.Group1.analog_vh1',1,0);
hr=mxOPC('setdoublecache','Channell.Device1.Group1.analog_izh1',1,0);
hr=mxOPC('Startdoublenotify','Channell.Device1.Group1.analog_vh1');
hr=mxOPC('Startdoublenotify','Channell.Device1.Group1.analog_izh1');
hr=mxOPC('readcache');
hr=mxOPC('writecache');
i=i+1;
end
```

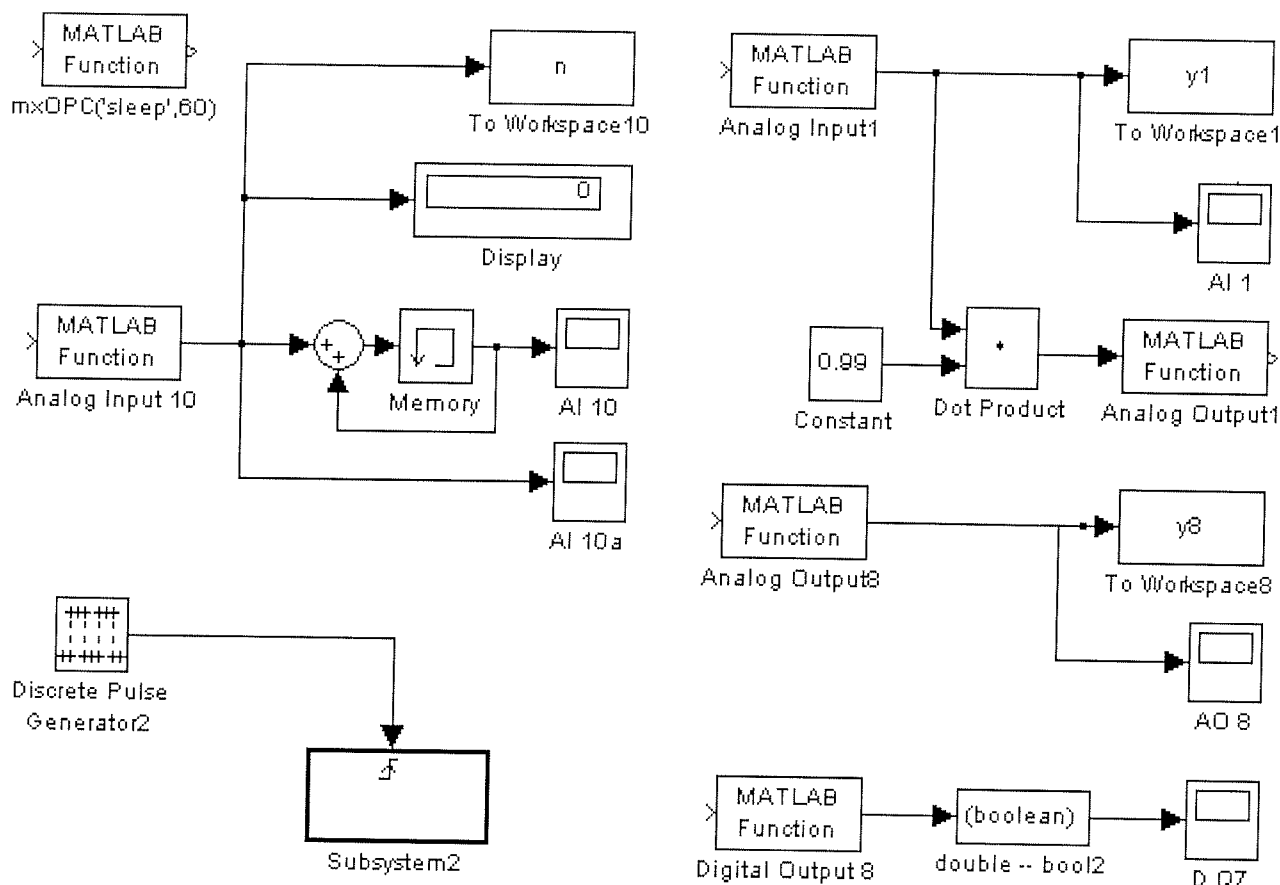
Slika 3: Primer programa v Matlabu za komunikacijo z enim vhodom in enim izhodom

Matlabovo okolje je možno sinhronizirati s procesom. S startno notifikacijo ('startnotification') in čakalnimi ukazi ('wait') je možno ustaviti Matlabovo okolje vse dokler podatek ni poslan na OPC strežnik. Druga pot za sinhronizacijo Matlabovega okolja je možna z uporabo sleep ukazov, kar je uporabljeno za testiranje hitrosti komunikacije in je prikazano na sliki 4.

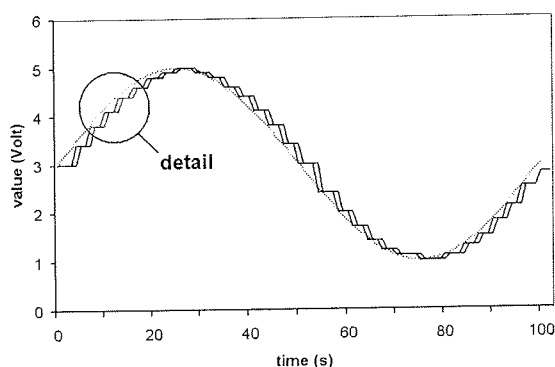
Za testiranje hitrosti prenosa podatkov iz realnega sveta v Matlab je bil uporabljen zvezen sinusni signal, prikazan na sliki 5 iz katere je razvidna diskretizacija kot posledica tipanja. Testiranja oziroma meritve smo izvajali v za sistem najbolj neugodnih razmerah, saj so se le tako lahko pridobili podatki o mejnih zmogljivostih. Strežnik je zajemal podatke iz krmilnika, preveril kvaliteto in morebitno spremembo vrednosti podatkov ter jih posredoval odjemalcu, ki jih je po

obdelavi poslal nazaj po isti poti. Za namene "worst case" testiranja se je uporabilo sinhrono branje, kjer se je vsak podatek vsakič spremenil, na istem računalniku pa so tekli en strežnik, dva odjemalca, SCADA ter Matlab program. Na sliki 6 je prikazan detajl signalov, kjer je z *a* označen sinusni signal, z *b* podatki na strežniku ter s *c* podatki na odjemalcu. Skupen čas prenosa je označen s t_3 , osnovo oziroma fiksni del časa prenosa predstavlja konstanta $t_1 = 300$ ms, kar je najmanjši čas kontinuiranega osveževanja uporabljenega OPC strežnika, razlika do izmerjenega časa pa je posledica obremenitve strežnika, ki je znašala povprečno $t_2 = 11$ ms za 11 signalov, torej okrog 1000 podatkov/s (vse na sliki 6). Omenjeni rezultati so v skladu s specifikacijami OPC standarda ter tudi skladni z opažanji drugih raziskovalcev /3//15//23/, ki so izvajali testiranja na industrijskih SCADA sistemih.

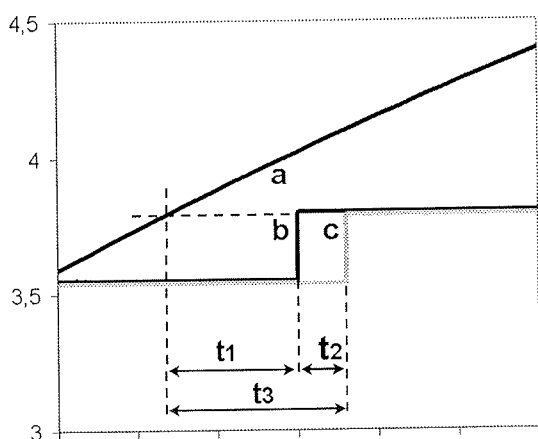
Drugi del testiranja se je nanašal na zanesljivost komunikacije. Opazovali smo razmerje med uspešno in neuspešno poslanimi podatki. Testiranje smo izvajali v zaprti zanki krmilnik-Matlab-krmilnik, ob čemer smo spreminjali tako čas "sleep" v Matlabu kot tudi obremenitev računalnika z drugimi opravili. Za kot najbolj neugodno se je izkazala manipulacija s slikami. V Matlabu smo v fiksni časovni intervali prožili cikle obdelave podatkov ter spremljali razmerje izvršenih ter neizvršenih ciklov. Če se cikel ni izvršil in je bil posledično prenos podatka neuspešen, smo sklepali na



Slika 4: Del testnega okolja v Matlabu



Slika 5: Signali za testiranje prenosov podatkov



Slika 6: Detajl signalov

zasedenost računalnika z drugimi opravili ter s tem na preobremenjenost. Ob tem naj poudarimo, da tudi klasični sistemi industrijske avtomatizacije nimajo 100% zanesljive komunikacije, ampak se izpadi podatkov pojavljajo tudi med normalnim delovanjem, kar se rešuje s ponovitvijo zahteve po prenosu izpadlega podatka (retry).

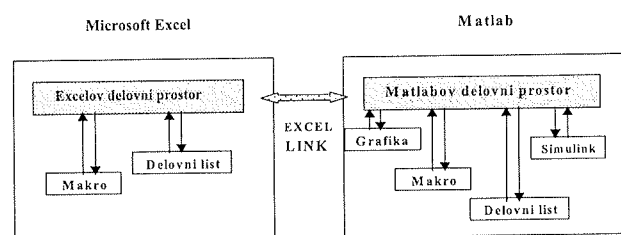
Testiranje smo izvedli pri časih proženja Matlabovih ciklov 20 ms in 60 ms pri različnih obremenitvah operacijskega sistema (OS), ter pri obdelavi enajstih realnih podatkov iz procesa. Rezultati so prikazani v tabeli 1, kažejo pa, da je pri časih izvrševanja 60 ms in več poraste zanesljivost na sprejemljivo vrednost tudi za industrijske razmere. Nekoliko zmanjšana hitrost ne predstavlja posebnih težav, saj je v industrijskih procesih večina prenosov podatkov do PC računalnika razreda sekund.

	Pročenje cikla v Matlabu	
	20 ms	60 ms
Neobremenjen OS	0,12 %	0,02 %
Obremenjen OS	1,69 %	0,61 %

Tabela 1: Rezultati testiranja izvajanja Matlab aplikacije – procent neizvršenih ciklov

Tretji del testiranja smo izvedli podobno kot v drugem primeru ob zaprti zanki krmilnik-Matlab-krmilnik, pri čemer smo ob znani hitrosti delovanja Matlaba merili skupen čas prenosa (t_3 na sliki 6), ki je povprečno znašal 720 ms, kar je še znotraj sprejemljivih mej za nadzor industrijskih procesov. Omenjen čas prenosa običajno ni sprejemljiv za vodenje procesov, čemur pa ne SCADA, ne opisana aplikacija nista namenjena, saj za to skrbijo krmilniki.

V četrtem delu smo poskušali poiskati alternativno pot za prenos industrijskih podatkov v Matlab. Obstaja možnost, da se podatki preko izbranega KEPServerEx vmesnika prenašajo v Microsoft Excel, kjer se da do njih on-line dostopati tudi iz Matlaba s pomočjo Excel linka, ki je podobno kot OPC odjemalec skupek nekaj dodatnih funkcij za Matlab. Sestavljena je bila aplikacija kot na sliki 7, izmerjene pa so bile hitrosti oziroma časi potrebni za prenos podatkov.



slika 7: Povezava Matlaba in Excela

Razvidno je bilo, da podatek prihaja iz krmilnika v Matlab hitreje, kot pa se izpisuje iz Matlaba na krmilnik. Razlog je v makru, ki vpisuje podatek na OPC strežnik oziroma na krmilnik. Opisana povezava je zelo občutljiva na obremenitev računalnika, saj so bili časi prenosa podatkov od 600 do 1200 ms, pri čemer naj omenimo, da je bil od obremenitve bolj odvisen čas branja podatkov v Matlab, kot pa čas vpisovanja podatkov v PLC.

Kot zadnje smo izvedli še test v pogojih, ki jim je predlagana rešitev najbolj namenjena. To je prenos procesnih podatkov v Matlab za namene simulacije, optimizacije in testiranja in sicer podatkov, ki jih tako ali tako že zajema SCADA sistem (slika 2). Uporabljena je bila torej standardna struktura sistema industrijske avtomatizacije, ki smo ji dodali povezavo do Matlaba. Pomembno je, da pri predstavljeni rešitvi ne prihaja do časovnih zakasnitev med podatki v Matlabu in tistimi v SCADA sistemu, saj sta oba priključena na isti OPC strežnik. Tudi morebitnih zakasnitev delovanja celotnega sistema zaradi dodatnega OPC Klienta ni bilo zaznati. Na procesih, ki imajo časovno konstanto večjo od ene minute ponuja predlagana rešitev tudi možnost vodenja brez omejitev, saj je v tem primeru čas tipanja 100 krat večji od časovne konstante in se lahko signal obravnava kot zvezen signal. Ta možnost se lahko izkoristi v učne, eksperimentalne ali raziskovalne namene, ob čemer ohranja vse prednosti uporabe zanesljivih in preverjenih industrijskih rešitev.

5. Zaključek

Namen OPC standarda je zagotoviti možnost medsebojnega povezovanja različne procesne opreme in programskih paketov različnih proizvajalcev ter s tem preprečiti, da bi bila programska oprema odvisna od strojne opreme proizvajalcev. Ugotovljeno je bilo, da prinaša uporaba te tehnologije številne prednosti, ki jih lahko strnemo v boljši povezljivosti z različnimi sistemi in procesnimi napravami ter boljši funkcionalnosti, ki jo OPC v povezavi s SCADA sistemi ponuja. Dogajanja na trgu SCADA programske opreme kažejo, da je praktična uporaba omenjene tehnologije dejansko trend razvoja na tem področju. SCADA sistema, kot sta IFIX ter InTouch, že temeljita na komponentni arhitekturi ter OPC in s tem tudi pričata, da je uporaba OPC realnost. Pri avtomatizaciji proizvodnje pomeni odločitev za OPC ne samo trenutno prednost zaradi primerljivosti različnih ponudb izvajalcev, ampak tudi ali pa predvsem odprtost sistema za nadaljno širitev saj so enkrat zajeti podatki resnično na voljo ostalim programom brez morebitnih posegov v SCADA sistem. Razvoj OPC tako ni samo pripomogel k večji izbiri proizvodov in konkurenčnosti, pač pa tudi k vedno večji uporabnosti in funkcionalnosti sistemov procesnega vodenja v industriji vse do sistemov poslovnega odločanja. Tudi na področju OPC gre razvoj v smeri povezovanja z Internetom, kjer so v fazi zaključnih dogovorov specifikacije za področje OPC XML ter nova specifikacija OPC Data Access 3.0.

Predlagana je rešitev, ki preko OPC uspešno poveže že obstoječ sistem industrijske avtomatizacije z okoljem Matlab, kar lahko služi za on-line spremljanje procesnih podatkov proizvodnje z namenom analize ali optimizacije proizvodnega postopka, zgodnje detekcije napak, ali pa zgolj prenosu procesnih podatkov v Matlab za nadaljnjo obdelavo ali shranjevanje. Omenjena rešitev pokaže svojo uporabnost še posebej v pogojih, ko se proizvodnega postopka ne sme spreminjati ali zaustaviti, kot je to primer v farmacevtski industriji ali v proizvodnji mikroelektronike, ko je tudi sama prisotnost ljudi v proizvodnji nezaželena. V testnem sistemu je OPC strežnik zajemal podatke iz krmilnika, preveril kvaliteto in morebitno spremembo vrednosti podatkov ter jih posredoval OPC odjemalcu, ki jih je po obdelavi poslal nazaj po isti poti. Za namene "worst case" testiranja se je uporabilo sinhrono branje, kjer se je vsak podatek vsakič spremenil, na istem računalniku pa so tekli en strežnik, dva odjemalca, SCADA ter Matlab program. Opravljene meritve kažejo, da so zaradi fiksnega mrtvega časa zmogljivosti najslabše pri prenosu enega samega podatka, pri velikem številu podatkov pa se lahko računa na hitrosti okrog 1000 podatkov/s. Procesni podatki so Matlabu na razpolago praktično istočasno kot SCADA sistemu, kar potrjuje uporabnost v industrijskih sistemih. Predlagana rešitev je še posebej primerna za počasi se spreminjajoče industrijske procese, kjer ponuja tudi možnost vodenja.

Možnosti nadaljnjih raziskav se kažejo v raziskavi uporabe OPC strežnikov z manjšimi časi prenosov podatkov v povezavi z opcijo "prenos podatka le ob spremembi" (exception based), kar pomeni v povprečju sicer krajši, vendar nederminističen čas prenosa. OPC omogoča tudi "time stamping", kjer se v hitrem krmilniku pripne podatku zraven njegove vrednosti še časovna znamka nastanka podatka. Ob prenosu takih podatkov v Matlab bi se le-ti lahko uskladili z ostalimi on-line podatki, kar prinaša nove izzive na področju vodenja procesov s spremenljivimi mrtvimi časi.

Literatura

- /1/ M. Kranjc, R. Karba. Mnenja o prihodnosti uporabe omrežnih tehnologij na področju nadzora in vodenja industrijskih procesov. B. Zajc in F. Solina, *Zbornik 8. Elektrotehniške in računalniške konference ERK'99*, str. 239-242, Portorož, 1998.
- /2/ P. Studebaker. ALL For One - The Top 10 Trends in Process Control. *Control Magazine* 04/01/1998.
- /3/ Al Chisholm, Intellution Inc. OPC/OLE for Process Control Overview. *World Batch Forum 1998*, www.intellution.com/opchub/opc_wbfoverview.asp
- /4/ M. Stianko, Merz. OPC Server According to the Data Access 1.0 and 2.0 Specification- Technical Descripton. *Czech Avtomatizace magazine No. 11/2000*, www.merz-sw.com/articles/opc_da.php3
- /5/ Al Chisholm., A Technical Overview of the OPC Data Access Interfaces, Intellution Inc., 1998
- /6/ Peršin S., Bratina B., Tovornik B., OPC vmesnik v sistemih procesnega vodenja, B. Zajc in F. Solina, *Zbornik 10. Elektrotehniške in računalniške konference ERK2001*, pp. 221-224, Portorož, 2001
- /7/ Pattle R., Ramisch J., OPC de facto standard for real time communication, *Parallel and Distributed Real-Time Systems*, Geneva, 1998
- /8/ Henry Je., Test case selection for simulations in the maintenance of real-time systems, *Journal of software maintenance research and practice*, 12/2000
- /9/ Mohamed Am et al., Real-time implementation of a robust H-infinity controller for a 2-DOF magnetic micro levitation positioner, *Journal of dynamic systems measurement and control transactions of the asme*, 12/1995
- /10/ Rodgerson J., Teaching systems and control using Matlab, *International journal of electrical engineering education*, Manchester Univ. Press, Manchester, 6/1992
- /11/ Zhi Wang, Research and implementation of fieldbus interoperability, *The 3rd world congress on Intelligent Control and Automation*, Hefei, China, 2000
- /12/ Janke M., OPC-plug and play integration to legacy systems, *Pulp and Paper Industry Technical Conference*, Atlanta, GA, USA, 2000
- /13/ Wu Sitao, Combining OPC with autonomous decentralized systems, *International Workshop on Autonomous Decentralized Systems*, Chengdu, China, 2000
- /14/ Wu Sitao, Using device driver software in SCADA systems, *Power Engineering Society Winter Meeting*, Singapore, 2000
- /15/ Mintchell GA, OPC integrates the factory floor, *Control Engineering, Channers-Denver Publishing Co. Oak Brook*, 1/2001
- /16/ Shimanuki Y., OLE for Process control (OPC) for new industrial automation systems, *IEEE International Conference on Systems, Man, and Cybernetics*, Tokyo, Japan, 1999

- /17/ Chaners-Denver Publishing: PC-based control software moves to VBA, NT, OPC, *Control Engineering*, Chaners-Denver Publishing Co., Oak Brook, 5/1998
- /18/ Linda Wills et al., An Open Platform for Recofigurable Control, *IEEE Control Systems Magazine*, vol. 21 no. 3, 2001
- /19/ Kirner T. G., Quality requirements for real-time safety-critical systems, *Control Engineering Practice*, vol.5, no. 7, pp. 965-973, 1997
- /20/ David W. Coit, Tongdam Jin, Prioritizing System-Reliability Prediction Improvements, *IEEE Transactions on Reliability*, vol. 50, no. 1, 2001
- /21/ Mitsugu Kishimoto, Optimized Operations by Extended X-Factor Theory Including Unit Hours Concept, *IEEE Transactions on semiconductor manufacturing*, vol.14, no. 3, 2001
- /22/ OPC Foudation, Data Access Automation Interface Standard 2.04, *OPC Foundation*, 2001
- /23/ Frank Iwanitz, Jurgen Lange, OLE for Process Control, *Huthig GmbH Heidelberg, Germany*, 2001

Stojan Peršin, univ. dipl. ing. el.
izr. prof. dr. Boris Tovornik,
doc. dr. Nenad Muškinja

vsi Univerza v Mariboru,
Fakulteta za elektrotehniko,
računalništvo in informatiko
2000 Maribor, Smetanova 17, Slovenija
e-mail: stojan.persin@uni-mb.si

Prispelo (Arrived): 10.12.2001

Sprejeto (Accepted): 25.04.2002