ISSN 1392 – 1215 ELEKTRONIKA IR ELEKTROTECHNIKA. 2003. Nr.4(46)

T 190 ELEKTROS INŽINERIJA

Objektų nuotolinio valdymo e.laboratorijoje aspektai

O. Ramašauskas, A. A. Bielskis
Klaipėdos universitetas, Informatikos katedra
Manto g. 84, LT-5800 Klaipėda

Įžanga

Klaipėdos universiteto Informatikos katedroje jau nebe pirmi metai dirbama prie e.mokymo idėjos. Dabar tai visame pasaulyje aktuali ir plačiai diskutuojama tema. E.laboratorija suvokiama kaip e.universiteto sudedamoji dalis – terpė, kur galima mokytis iš nutolusių darbo vietų, dėstytojo padedamam arba savarankiškai skaitant elektroninius konspektus, pasitikrinant žinias [9].

Praktika rodo, kad geriausiai įsisavinamos ir išlaikomos tos žinios, kurios pagrįstos ir įtvirtintos asmenine darbo patirtimi. Tokios patirties šaltinis ir galėtų būti e.laboratorijoje atliekami teorinio bei praktinio pobūdžio eksperimentai [6]. Čia turėtų būti galimybė juos ne tik savarankiškai atlikti, bet ir teoriškai analizuoti gautus rezultatus, elektroninėmis komunikacijos priemonėmis konsultuotis su kolegomis, peržiūrėti seniau atliktus panašius darbus, palyginti rezultatus [7].

Manoma, kad, augant studentų skaičiui, įprastinio mokymo efektyvumas ims mažėti, nes dėstytojai nebegalės paskirstyti dėmesio ir laiko daugybei studentų. Didės e.mokymo, nuotolinio įvertinimo svarba, adaptyvių agentinių technologijų įtaka [1].

Darbo tikslai ir uždaviniai

Svarbiausia e.laboratorijos funkcionavimo sąlyga – suteikti galimybę atlikti mokslinį eksperimentą ir parengti ataskaitą nuotoliniu būdu. Tai užtikrintų projektuojamoji tinklinė (agentinė) *eksperimento nuotolinio valdymo sistema* (ENVS) su *nuotolinio valdymo sąsaja* (NVS). Įgyvendinus projektą, atsivertų naujos e.laboratorijos koncepcijos plėtojimo galimybės [3, 4], pavyzdžiui, sukurti virtualios realybės teatrą e.gamybai vizualizuoti [7].

Atlikus išankstinę uždavinio analizę ir numačius galimus sprendimo variantus, buvo užsibrėžti šie darbo tikslai:

- sukurti eksperimentams atlikti tinkamą koncepcinį e.laboratorijos modelį;
- parinkti modeliui įgyvendinti tinkamus įrankius ir priemones (programinius agentus);
- sukurti kompiuterizuotus laboratorinius stendus ir/arba programinius jų modelius (algoritmus);
- sukurti jų vietinio ir nuotolinio programinio valdymo priemones (algoritmus, programas);
- sukurti vartotojo aplinką laboratoriniams darbams;
- suprojektuoti specializuotą tinklinę DB vartotojams ir jų veiklai registruoti, duomenims apie modelius ir laboratorinių darbų rezultatams kaupti;
- sukurti sistemos demonstracinę versiją su veikiančiais modeliais.

E.laboratorijos modelio konceptai

1. Neformalieji reikalavimai. E.laboratorijos eksperimentų nuotolinio valdymo sistemos koncepcinis modelis bei pati sistema turi tenkinti paskirstytojo naudojimo trijų lygių koncepciją.
   1.1. Prezentacijų lygį turi sudaryti vartotojo aplinka ir pagalbos sistema.
   1.2. Nuotolinio eksperimento vykdymo lygį turi sudaryti:
      1.2.1. kompiuterizuoti laboratoriniai stendai ir/arba jų modeliai;
      1.2.2. jų vietinio valdymo priemonės;
      1.2.3. jų nuotolinio valdymo priemonės.
   1.3. Duomenų bazės lygį turi sudaryti specializuota tinklinė duomenų bazė.
2. Lygiai turi būti tarpusavyje suderinti.
3. Vartotojo aplinkos reikalavimai.
   3.1. Aiški struktūra ir nesudėtingas valdymas.
   3.2. Galimybė identifikuoti vartotoją sistemoje ir suteikti jam paslaugas, atitinkančias jo teises.
   3.3. Paprasta navigacija, suteikianti galimybę bet kada pereiti į norimą dalį ir greitai atlikti užduotį.
   3.4. Spartus keitimasis informacija su tinkline DB.
4. Reikalavimai stendų ir/arba modelių programoms.
   4.1. Gebėti parinkti valdymo parametrus.
   4.2. Nustatyti atliekamo eksperimento rezultatus.
5. Reikalavimai vietinio stendų ir/arba modelių valdymo priemonėms nuotolinio ryšio terpėje.
   5.1. Gebėti parinkti valdymo parametrus bei nustatyti rezultatus.
   5.2. Gebėti bendrauti su nuotolinio valdymo sąsaja.
6. Nuotolinio valdymo sąsajos funkciniai reikalavimai.
   6.1. Identifikuoti užsakytą eksperimentą.
   6.2. Valdyti stendą/modelį pagal užsakyto eksperimento parametrus.
   6.3. Priimti ir kaupti modelio darbo rezultatus DB.
7. Reikalavimai specializuotai tinklinei duomenų bazei.
   7.1. Reikalavimai DBVS:
      7.1.1. nedidelė įrangos apimtis;
      7.1.2. nedidelių kompiuterio išteklių naudojimas;

7.1.3. nesudėtingas administravimas;
7.1.4. nemokama.

7.2. Reikalavimai projekto DB, kurioje saugoma:
7.2.1. vartotojų prisijungimo vardai, slaptažodžiai, teisės, kita informacija;
7.2.2. vartotojų teisių lygiai bei privilegijos;
7.2.3. eksperimentų: numeris, būsena, kuriuo modeliu atliekamas, atlikimo laikas;
7.2.4. modelių: įsijungimo numeris, įėjimo (išėjimo) vertės, pavadinimai, aprašymai;
7.2.5. eksperimentų pradiniai duomenys;
7.2.6. eksperimentų rezultatai, ataskaitos.

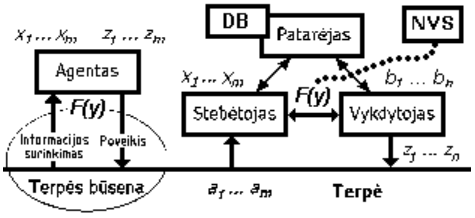
8. Suderinamumo reikalavimai.
8.1. Specializuotos tinklinės DB platforma turi būti suderinama su NVS ryšio terpe.
8.2. Vartotojo aplinkos terpė turi būti suderinama su specializuotos tinklinės DB terpe.

E.laboratorijos tinklinė struktūra

E.laboratorijos veikimą gana tiksliai apibūdina jos terpėje veikiančių programinių agentų funkcijų visuma ir sprendžiamų uždavinių santykis. Panagrinėkime vartotojo požiūriu svarbius e.laboratorijos valdymo, sąsajos ir eksperimento realizavimo aspektus. Įrodyta, kad ekspertinės, agentinės modeliavimo ir tyrimo priemonės naudingos ir moksliniams tyrimams e.laboratorijoje, ir e.mokymuisi [8].

Terpės poveikių aibę galima apčiuopti jutiklių x_i signalais $\{x_1 \dots x_m\} \in X$, kuriuos atitinka poveikių verčių sąrašai $\{a_1 \dots a_m\} \in A$. Aibę X galima modeliuoti programa, pvz., atsitiktinių skaičių generatoriumi; čia $\{\forall x \in X|A\}$. Analogiškai, jeigu kiekvieną valdiklį b_j sutapdinsime su atitinkamu įtaiso su perdavimo funkcija y, kurio reakcijų sąrašas $\{z_1 \dots z_n\} \in Z$, išėjimo kintamuoju, tai kiekvienam modeliuojam įtaisui $\{\forall z \in Z|B \Rightarrow A(y)\}$.

E.laboratorijoje seką $A \rightarrow F(y) \rightarrow Z$ vykdantys procesai yra jos intelektualūs agentai, kurių akcijos/reakcijos priklauso nuo sudaryto valdymo algoritmo ir DB įrašytų žinių (1 pav.). Agentas – tai savarankiška, susivokianti aplinkoje programa, galinti valdyti savo veiksmus, sekdama vieną ar daugiau objektų. Agentas gali turėti mokymosi (gebėjimo gerinti darbą, adaptyvumo) atributą.

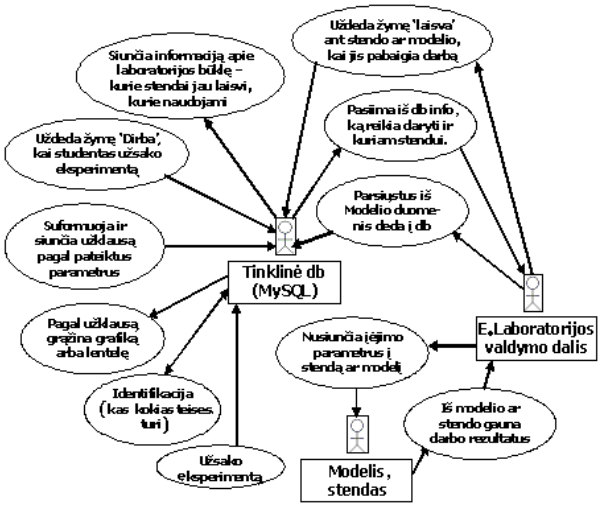


1 pav. Sutelktojo ir paskirstytojo agentų modeliai žinių terpėje

Jeigu agentas sąveikauja ne su pačia terpe, bet su jos lingvistiniu aprašymu, tai jis turi suprasti formalią terpės aprašymo kalbos abėcėlę, gramatiką ir mokėti ja naudotis (suprasti ir vertinti būsenas). Skirtingai nuo žmogaus intelektui būdingų samprotavimų, agentas vadovaujasi griežtai formalizuota logistika. Artinant agento „mąstymą“ prie žmogiškojo, naudotina neraiškiosios logikos metodologija (susiejant jutiklių $\{x_1 \dots x_m\}$ ir programuojamų valdiklių $\{z_1 \dots z_m\}$ signalų apdorojimą pagal IEC 1131 rekomendacijas) [2]. Didelėse, atvirose, dinamiškose ir neprognozuojamose terpėse geriausiai dirba daugiaagentės sistemos (MAS, MultiAgent Systems), kuriose svarbu nustatyti agentų mokymosi strategiją, leidžiančią jiems adaptuotis dinamiškai kintančioje terpėje.

Eksperimento nuotolinio valdymo modelis

Vienas svarbiausių programinių e.laboratorijos komponentų yra eksperimento nuotolinio valdymo sistema su nuotolinio valdymo sąsaja. Formuodami nuotolinio valdymo sistemos išplėstinę objektinę struktūrą, išskiriame ENVS sąsajas su modeliais ir tinkline DB (2 pav.).



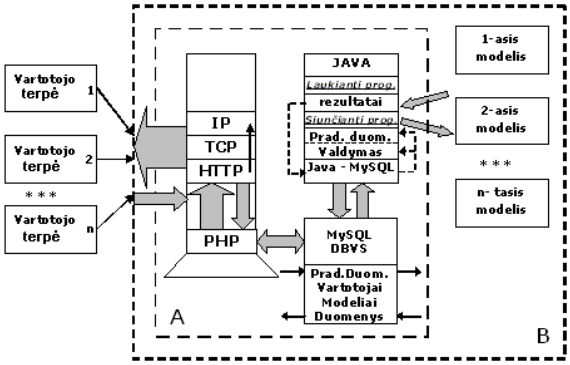
2 pav. ENVS pagrindinių sąsajų diagrama [4]

Kuriami objektai standartine notacija yra vaizduojami trijų blokų schema (1 lent.). Pirmame bloke nurodytas objekto pavadinimas, antrame – objekto savybės, o trečiame – tam objektui būdingi metodai.

Suderinus konceptus su numatytomis įgyvendinimo priemonėmis, pridėjus papildomus reikalavimus, gauta bendroji ENVS struktūra (3 pav.). Čia objektams valdyti iš nutolusios sistemos vartotojui siūloma naudoti HTTP protokolą, veikiantį TCP/IP protokolo pagrindu. Jis leidžia sukurti objektams valdyti tinkamą vartotojo sąsają, perduoti informaciją tekstiniu, skaitiniu ir grafiniu pavidalu. Duomenims perduoti vartotojui ir grąžinti atgal sistemai galima naudoti nemažai serverių, pavyzdžiui, Microsoft IIS ir Apache HTTP. IIS šiek tiek paprastesnis už Apache, tačiau yra apmokamas ir veikia tik Windows OS, todėl siūlomas naudoti Apache [0] su PHP preprocesoriumi (phpMyAdmin). Techniškai nuotolinio valdymo sąsają (A) sudaro trys pagrindinės komponentės – Apache HTTP serveris su PHP preprocesoriumi, MySQL DB ir Java programos, aptarnaujančios nutolusius stendus. Vartotojas mato PHP scenarijaus dinamiškai sugeneruotą puslapį, kuriame arba vaizduojami užklausomis iš duomenų bazės paimti duomenys, arba forma, kurią užpildžius ir pasiuntus duomenis į serverį, jie bus padėti į duomenų bazę. Čia juos galės pasiekti Java programa, siunčianti komandas ir duomenis stendams.

1 lentelė. Iš koncepcinio modelio apibrėžti penkių tipų objektai

1. Vartotojas	Vardas, pavardė, grupė, šaukinys, slaptažodis, teisės	Identifikuoti sistemą, pasirinkti modelį, nustatyti parametrus, peržiūrėti rezultatus, analizuoti duomenis
2. Vartotojo terpė	Tinklo adresas, SQL užklausos CSS dalis	Identifikuoti vartotoją, patvirtinti atributus, siųsti užklausas DB, užsakyti eksperimentą, dinamiškai pateikti sąsajos puslapius (*modelių sąrašą*, *modelio parametrus*, *eksperimentą*, *eksperimento duomenis*: *grafiką ar lentelę, pastabas*)
3. Tinklinė DB	Vartotojai, teisės, modelių sąrašas, modelių darbo rezultatai, eksperimentų sąrašas, modelių darbo parametrai, pastabos	Grąžinti užklausų rezultatus
4. Nuotolinio valdymo sąsaja (NVS)	Lentelių pavadinimai, duomenų perdavimo protokolas	Pažymėti modelį, imti duomenis iš DB, siųsti duomenis į modelį, imti duomenis iš modelio, saugoti duomenis DB, tikrinti eksperimento būseną
5. Stendas (modelis)	Darbinio kanalo # ir protokolas, priėmimo kanalo # ir protokolas, ID	Imti pradines reikšmes, atlikti eksperimentą, koduoti duomenis, išsiųsti duomenis į NVS



3 pav. Vartotojo, NVS (A) ir ENVS (B) sąveikos schema

NVS Java programos su vartotoju nesusijusios, pagrindinis informacijos šaltinis yra DB lentelės, kurių turinys nuolat tikrinamas. Atitinkamų „žymėtų“ laukų verčių eksperimentų lentelėje pasikeitimas į „0“ arba įrašų su šia žyme atsiradimas interpretuojamas kaip pradinių duomenų nustatymas konkrečiam stendui. Iš eksperimento imamas jo identifikacinis numeris (ID) ir iš pradinių duomenų lentelės (unikalios kiekvienam stendui) imami pradiniai duomenys (modeliavimo ar eksperimento parametrai). Modelis, gavęs pradinius duomenis, grąžina darbo rezultatus (pasiunčia juos laukiančiai Java programai – serverio daliai), kuri juos surašo į duomenų bazę. Jei modelis dirba lėtai, atnaujindamas duomenis, PHP preprocesorius pokyčius gali parodyti lentelėje ar grafiškai.

Nuotolinio valdymo ryšio terpės sąsajos analizė

Siūloma TCP/IP protokolo pagrindu veikiančio tinklo terpė su lizdų (socket) mechanizmu. Jos visiškai pakanka šiame etape užsibrėžtiems tikslams, nes:

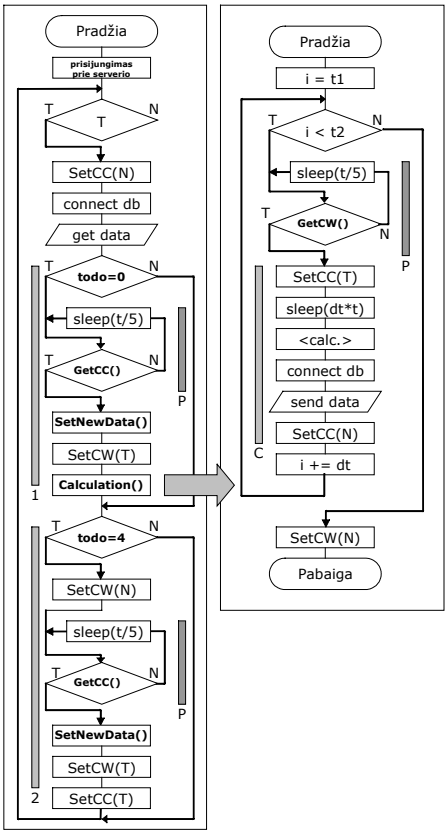
- galima perduoti duomenis skaitmeniniu pavidalu (5.2 reikalavimas);
- yra pritaikyta MySQL (8.1 reikalavimas);
- nereikia kurti unikalios duomenų perdavimo terpės.

Protokolo pasirinkimas. UDP transporto lygiu perduoda duomenis kaip žinių paketus, be klaidų korekcijos. Nagrinėjamu atveju klaidos neleistinos, todėl pasirinktas TCP protokolas, kuris transporto lygiu perduoda duomenų segmentų srautą, automatiškai ištaisydamas klaidas.

Vietinio valdymo programa stendui aptarnauti

Java kalba parašytos programos gali veikti tolimuose kompiuteriuose ir valdyti prijungtus fizinius arba virtualius laboratorinius stendus (programinius modelius) [0]. Kadangi stendui arba modeliui valdyti naudojamas tas pats algoritmas, duomenys perduodami tuo pačiu formatu, analogiška eilės tvarka, dėl to valdymui tinka ta pati programa, pakeitus kai kuriuos konkretaus modelio parametrus, konstantas. Ją sudaro du pagrindiniai moduliai (4 pav.):

- ryšio su nuotolinio valdymo priemonėmis agentas;
- programuojamo stendo valdymo agentas.



4 pav. Ryšio ir stendo programinio valdymo moduliai

Stendo valdymo dalis gali varijuoti, jeigu valdomas objektas yra fizinis laboratorinis stendas arba jo veikimą imituojantis programinis modelis. Realiame fiziniame stende verčių skaičiavimų funkciją galėtų pakeisti stendo elektrinių ir kt. parametrų matavimo funkcija. Čia teikiamas stendo veikimą imituojantis programinis modelis.

Valdymo dalies algoritmas gana trivialus. Praktiškai jis susideda iš keleto dalių: iš pagrindinės programos objekto „modelis“ kaip atskiras procesas iškviestas metodas „calculation()“ pagal pradinius kintamuosius pradeda vykdyti ciklą – keičia kintamąjį „i“, pradėdamas pradiniu laiko momentu „t_1“ ir baigdamas galutiniu laiko momentu „t_2“. Funkcija „getCW()“ patikrinama kintamojo – „žymės“ vertė (ar nėra stabdomi skaičiavimai). Jei stabdomi, tai laukiama, kol sustabdęs skaičiavimus pagrindinis procesas leis modeliui toliau veikti. Žymei „calcWorking“ esant ‘true’ (teigiamai), tai interpretuojama kaip leidimas skaičiuoti ir kintamojo „calcContinue“ vertė pakeičiama į ‘true’ (metodas „setCC(T)“).

Ryšio su serveriu algoritmas

Ryšio su nuotolinio valdymo priemonėmis agentas dirba jau nagrinėto TCP serverio programos algoritmu, tiktai operuojama su daugiau duomenų, o užuot išvedus į ekraną, duomenys analizuojami ir priklausomai nuo atėjusios komandos arba pradedamas darbas, arba keičiami darbo parametrai, arba modelis stabdomas.

Kol neatsiųsti duomenys iš nuotolinio valdymo priemonių, žymė „calcContinue“ keičiama į „false“ (metodas „setCC(N)“), neleidžiama pradėti skaičiavimų be pradinių duomenų. Kai tik duomenys ateina, analizuojama komanda „todo“. Jei tai „0“, vadinasi, reikia pradėti skaičiavimus, jei tai „3“, reikia pakeisti darbo parametrus, o jei tai „4“, – stabdyti modelio darbą (tai daroma pakeičiant parametrą „t_2“ į parametrą „i“ (metodas „setStopParam()“). Ciklas baigiasi.

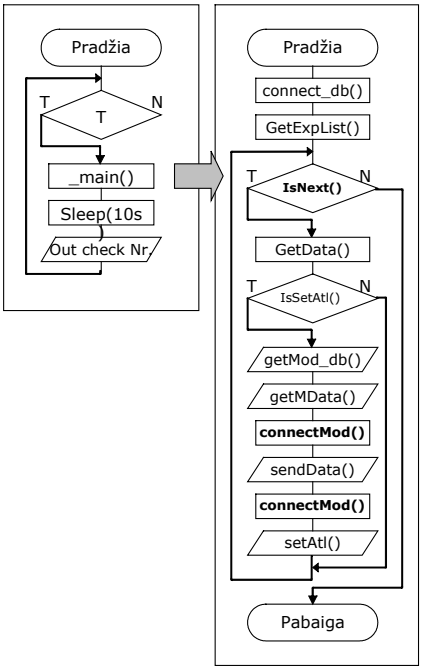
Parametrams keisti ši operacija būtų analogiška ir algoritmo schemoje nepavaizduota. Keičiasi tik sąlygos operatoriaus „todo“ vertė – ji turi būti lygi „3“. Jei modelis neveikia ir ateina pradiniai duomenys su „todo“ verte „0“, tai tikrinama, ar kas neskaičiuojama (metodas „getCC()“). Nustatomi modeliavimo parametrai (metodas „SetNewData()“), žymė „calcWorking“ pakeičiama į „true“ (metodas „setCW(T)“) bei paleidžiama jau nustatyta modelio valdymo dalis. Jei „todo“ vertė lygi „3“, tai pirmiausia „calcWorking“ prilyginama „false“ (metodas „setCW(N)“), darbas stabdomas ir laukiama, kol baigsis eilinis skaičiavimas (metodas „getCC()“). Keičiami parametrai (metodas „SetNewData()“), žymė „calcWorking“ pakeičiama į „true“ (metodas „setCW(T)“), t.y. modeliui leidžiama dirbti su naujais parametrais.

Nuotolinio valdymo algoritmas

Nuotolinį objektų valdymą atlieka dvi atskiros Java programos – agentai, naudojančios lizdų (socket) mechanizmą keistis duomenimis (5 pav.). Pirmoji siunčia modeliams pradinius duomenis bei komandas. Ją sudaro du procesai – pagrindinis (funkcija „main()“), begaliniame cikle kviečiantis pagalbinį (funkcija “_main()”), kuris minimaliu prioritetu tikrina DB lentelėje eksperimentų žymes.

Paleidusi šalutinį procesą, pagrindinė funkcija sustabdoma 10 sekundžių. Pagalbinis procesas veikia minimaliu prioritetu ir, neapkraudamas sistemos, jungiasi prie duomenų bazės. Metodu „connect_db()“ prisijungus prie duomenų bazės, peržiūrima visa eksperimentų lentelė. Tikrinamas kiekvienas eksperimentas, kol baigiasi visas eksperimentų sąrašas. Toliau skaitomi kiekvieno eksperimento duomenys metodu „getData()“. Jeigu eksperimento „žymė“ yra vykdomoji, tai „getMod_db()“ pasiima modelio parametrus.

Pagal „ID“ siunčiama užklausa į modelio pradinių duomenų lentelę ir imami pradiniai eksperimento duomenys (metodas „getMData()“), paskui siunčiami į prijungtą modelį „connectMod()“ → „sendData()“.



5 pav. Nuotolinio objektų valdymo algoritmas

Tai atlikus, darbas baigiamas ir pagalbinis procesas vėl paleidžiamas iškvietus jį iš pagrindinės programos. Ši rutina kartojama kas 10 sekundžių. Išsiuntęs duomenis agentas priima, apdoroja ir siunčia modelio darbo rezultatus programai, priimančiai ir saugančiai juos DB.

Rezultatų priėmimo ir saugojimo DB agentas

Pagal algoritmo struktūrą tai analogiška TCP Server programa, tik vietoj gautų duomenų išvedimo į ekraną formuojama užklausa ir duomenys surašomi į DB. Naudojami analogiški metodai kaip ir jau nagrinėtoje programoje, kinta tik siuntėjo ar gavėjo adresas, kiti neesminiai parametrai.

Tiek rezultatus siunčianti, tiek juos priimanti programos vartoja palyginti mažai serverio išteklių. Dėl to abi programos gali veikti tame pačiame kompiuteryje, kartu su MySQL DBVS, Apache HTTP serveriu, PHP interpretatoriumi ir kita reikalinga programine įranga [0].

Fizinis objektas e.laboratorijoje

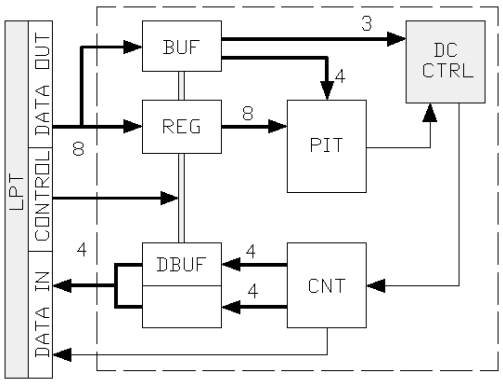
Panagrinėkime, kaip e.laboratorijoje sukuriami valdomi objektai ir kaip užtikrinama jų tarpusavio sąveika. Tarkime, valdomu objektu pasirinktas nuolatinės srovės kolektorinis mikrovariklis. Kompiuteriu valdysime sukimosi dažnį ir kryptį. Mikrovariklis turi grįžtamąjį ryšį su kompiuteriu – serveriu [0], prie kurio yra prijungtas per LPT uostą. Kadangi uosto signalų galia nedidelė, įdiegiama plokštė, skirta mikrovarikliui valdyti bei suderinti su kompiuteriu.

Mikrovariklis ir plokštė kartu toliau bus vadinami tiriamuoju valdomu objektu arba tiesiog objektu [1]. Šiam objektui apibrėžiami du reikalavimai: sukimosi krypties ir dažnio valdymas.

Pasinaudodami variklio rotoriaus inertiškumu, galėtume valdyti jo sukimosi dažnį, paduodami „0“ ir „1“ sekas [8], kurių trukmės varijuotų viena kitos atžvilgiu, todėl vidutinė įtampa apvijoje taip pat varijuotų. Tokia loginių lygių trukmės variacija gali būti gauta programiškai moduliuojant impulso trukmę (PWM, Pulse Width Modulation) ir leidžia valdyti objektą.

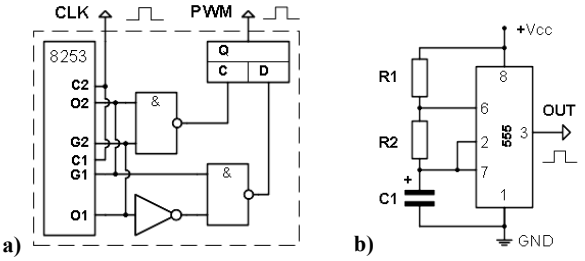
Tiriamojo objekto sandara ir valdymo algoritmai

Objekto struktūrinėje schemoje (6 pav.) esantis 8253 laikmatis (PIT) vaidina vieną iš pagrindinių funkcijų, nes leidžia reguliuoti variklio greitį. Variklio krypties pasirinkimą vykdo krypties pasirinkimo blokas DC_CTRL. Laikmačiui 8253 programuoti reikia 12 bitų, iš kurių 8 skirti duomenims, o 4 – operacijoms. Krypties blokui reikia dar 3 bitų, kurių 2 naudojami krypčiai pasirinkti ir 1 – krypčiai patvirtinti. Lygiagretusis (LPT) uostas teikia tik 8 duomenų bitus ir dar 4 kontrolinius. Be pagalbinės logikos abiejų blokų programuoti būtų neįmanoma. Pagalbinės logikos vaidmenį objekto schemoje atlieka 74374 registras (REG) ir buferinis magistralinis formuotuvas 8286 (BUF). Lustai 8286 ir 74374 valdomi dviem (iš 4) kontroliniais bitais.



6 pav. Valdomo objekto struktūrinė schema

Likusios dalys – skaičiavimo blokas (CNT) ir dvigubas buferis (DBUF, du 8286) – reikalingos sukimosi dažnio skaičiavimo blokui. Impulso trukmei nustatyti naudosime programuojamo intervalų laikmačio (8253) 5-ąjį režimą. Taip suprojektuotas PWM blokas parodytas 7 pav., a. Čia skaičiuotuvas yra perkraunamas per G išvadą, kurio loginis lygis tuo metu turi būti lygus „0“. Be to, esant tam pačiam režimui, skaičiuotuvo dekrementavimo pabaigoje išėjime O susiformuoja loginis „0“, lygus vienai C (CLK) impulso trukmei. Taigi, sujungus abiejų skaičiuotuvų išvadus O1 su G2 ir O2 su G1, skaičiavimo pabaigoje jie vienas kitą perkrautų.



7 pav. PIT: a) PWM blokas iš 8253, b) taktų generatorius

Skaičiavimas nesiliautų, kol nebūtų sustabdytas programiškai, o kiekvieno skaičiuotuvo skaičiavimo trukmė priklausytų nuo įrašyto pradinio skaičiaus į kiekvieną iš jų. Kadangi kiekviename iš išėjimų O skaičiavimo pabaigoje susiformuoja loginis „0“, o trigeriui, kad užfiksuotų duomenis, reikia „1“, tai abu PIT išėjimai prijungti prie aukščiau esančio IR-NE elemento 7400, kur tokiu būdu kiekvieno skaičiuotuvo skaičiavimo pabaigoje trigerio išvade C gaunamas impulsas, lygus vieno CLK impulso trukmei. Priklausomai nuo dydžių, įrašytų į skaičiuotuvus, tarpas tarp į trigerį paduodamų impulsų priklauso nuo tų dydžių, o C trukmė lieka pastovi. Taigi trigeris išlaikys duomenis tiek, kiek reikės. Svarbu, kad vienas iš impulsų formuotų „1“, kitas – „0“ trigerio D įėjime. Kadangi trigerio išėjimo Q būsena priklauso ir nuo D būsenos C impulso metu, todėl reikia suformuoti teisingą D signalą. O1 ir O2 PIT išėjimai gali įgyti tik tris reikšmes: 01, 10, 11 (5-uoju režimu 00 nesuformuojami). PIT veikimo metu pirmosios dvi kombinacijos atsiranda skaičiuotuvų skaičiavimo pabaigose, trys likusios – skaičiavimo metu. Sudėję reikiamą signalą gautume D išvade.

Stačiakampių impulsų generatoriaus (7 pav., b) impulsų dažnis apytiksliai apskaičiuojamas pagal formulę

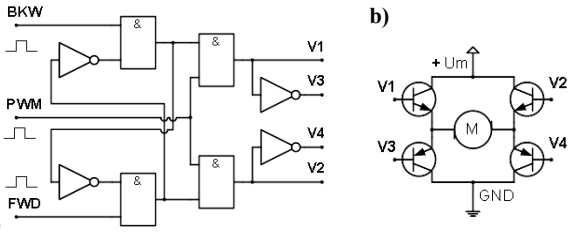
$$f = \frac{1}{0.693 \cdot C1 \cdot (R1 + 2 \cdot R2)}. \qquad (1)$$

Impulsų amplitudė yra beveik lygi maitinimo įtampai, o loginių „1“ ir „0“ trukmės gali būti nustatytos. Tarkime, kad t_1 ir t_0 yra loginių „1“ ir „0“ trukmių. Akivaizdu, kad $T = t_0 + t_1$, todėl abiejų loginių lygių trukmės bus atitinkamai lygios

$$\begin{cases} t_0 = 0.693 \cdot (R1 + R2) \cdot C1; \\ t_1 = 0.693 \cdot R2 \cdot C1. \end{cases} \qquad (2)$$

Aprašomame objekte reikalingas reversinis variklio srovės stiprintuvas (8 pav., b). Tokia schema galėtų būti valdoma tiesiai iš kompiuterio LPT uosto loginių elementų, prijungus valdymo signalus prie atitinkamų tranzistorių bazių. V1 ir V2 atsidarys esant loginiam „1“, o V3 ir V4 – loginiam „0“. Vengiant trumpojo jungimo,

jeigu vienu metu būtų įjungti V1 ir V3 arba V2 ir V4, sukurta loginė prevencinė schema, atmetanti išimtinių atvejų pasirodymą (8 pav., a; 2 lent.). Prevencinę funkciją atlieka 4 pirmieji loginiai elementai (2 NE, 2 IR). Tuo atveju, kai visi įėjimai įgyja loginį „1“, pirmenybė bus suteikta anksčiausiam.



a)

8 pav. Komutavimo logika (a) ir reversinis stiprintuvas (b)

2 lentelė. Išimtinių atvejų prevencinės schemos logika

BKW	FWD	PWM	V1	V4	V2	V3
0	0	0	0	1	0	1
0	0	1	0	1	0	1
0	1	0	0	1	0	1
0	1	1	0	1	1	0
1	0	0	0	1	0	1
1	0	1	1	0	0	1
1	1	0	0	1	0	1
1	1	1	1(0)	0(1)	0(1)	1(0)

Objekto valdymo tvarkyklė

Tvarkyklę patogu integruoti į pačią programą ir taip sudaryti bendrą sistemą – tvarkyklės ir bendravimo su vartotoju sąsają. Šis sprendimas ne tik palengvina kurti tokią sistemą, bet ir veikia daug greičiau, nes sutrumpėja transakcijų trukmė.

Tvarkyklei kurti pasirinkta objektinė Delphi kalba. Šią programavimo kalbą suderinus su žemo lygio kalba – asembleriu, buvo gautas patogus ir gana greitai veikiantis objekto valdymo skydelis. Čia sukimosi dažnio skaičiavimas yra pavestas programai. Tai turi tiek teigiamų, tiek neigiamų ypatumų. Nekurdami aparatinio skaičiuotuvo, eliminuojame papildomus testavimo ir suderinimo darbus. Perleisdami programai skaičiavimą, galime nesunkiai sukurti įvairius programinius filtrus. Objekto elgesys tiesiogiai priklauso nuo to, ką jam perduoda programa, todėl tiesioginė kreiptis į lygiagretų uostą yra būtina – tai gali sukelti keblumų Windows NT tipo operacinėse sistemose.

Programos ypatumas yra tas, kad valdymas ir patikra atliekami išoriškai. Darbo ataskaita saugoma tekstinėse rinkmenose, kurių pavadinimai unikalūs (pavadinimus sudaro keturių ženklų metai ir dviženkliai mėnuo, diena ir valanda). Praėjus valandai, sukuriama nauja tuščia rinkmena; į ją įrašoma 3600 verčių, nes rašoma kas sekundę. Tokie duomenys prieinami ENVS ir gali būti analizuojami eksperimento planavimo teorijos priemonėmis.

Įvertinimo ir tikslumo padidinimo perspektyva

Aprašytame pavyzdyje svarbu ne tik valdyti prie kompiuterio prijungtą objektą, bet ir stebėti jo būsenas, garantuoti grįžtamąjį objekto ryšį su vartotoju. Galima naudoti standartinius jutiklius arba susikurti aprašytus [3], pritaikyti nesudėtingus neraiškiosios logikos ir neuroninių tinklų modelius signalams apdoroti [8].

Matyti, kad ENVS gali ne tik tiesiogiai valdyti fizinį ar virtualų eksperimentą, bet iš gautųjų duomenų aprašymų atlikti charakteristikų aproksimavimą, tikimybinį apdorojimą, įvertinti imčių hipotezes, rangus ir kt. [5], naudodama standartinius programuojamus matematinius paketus (Maple, MatLAB, etc) arba originalias programas, kurias nesunku parašyti Java, Delphi ir kitomis objektinio programavimo kalbomis.

Išvados ir tolesnio darbo perspektyvos

Įgyvendinant šį projektą sukurta sistema sudaro sąlygas Klaipėdos universiteto e.laboratorijoje įrengti eksperimentų nuotolinio valdymo sistemą, leidžiančią atlikti tiek įvairius modeliavimo eksperimentus, tiek fizinių objektų tyrimus tinklinėje terpėje.

Sukurtasis supaprastintas koncepcinis e.laboratorijos modelis tinka nesudėtingiems nuotoliniams eksperimentams atlikti ir demonstruoti:

- leidžia sukurti kompiuterinius modelius, kurie imituotų fizinius laboratorinius stendus;
- paruošti jų vietinio bei nuotolinio programinio valdymo priemones;
- prisijungti tiek vietiniais, tiek plačiosios aprėpties tinklais.

Toliau plėtojant e.laboratorijos koncepciją ir plečiant eksperimentų dalį, reikėtų:

- detalizuoti koncepcinį e.laboratorijos nuotolinio valdymo sistemos modelį, taikant adaptyvias intelektualių programinių agentų technologijas;
- integruoti į vartotojo aplinką įvairių mokymo kursų e.konspektus, e.testus;
- pritaikyti modelius valdančias programas realiems laboratoriniams stendams valdyti;
- projektuoti ir įdiegti tikimybines ir neraiškiosios logikos aplikacijas veikiančių objektų modelių valdymo stabilumui ir tikslumui gerinti.

Literatūra

1. **Aylett R., Brazier F., Jennings N., Luck M., Preist C., Haynes H**. Agent Systems and Applications // The Knowledge Engineering Review, 13(3). – 1998. – P. 303–308.
2. **IEC 1131 – 7**: **Fuzzy Control programming**. – Commitees Draft. – TC65/WG, 7/TF8: CD1, 1997.01.19.
3. **Bielskis A**. **A**., **Moždžer A**. Objekto automatinis valdymas laboratorijoje // Vadyba. VLVK mokslo tiriamieji darbai: 2002 m. pr. Nr.1. – Klaipėda: KU leid., 2002. – P. 48–56.
4. **Bielskis A**. **A**., **Narbutas M**. Nuotolinio valdymo sąsaja e.laboratorijoje // Vadyba. VLVK mokslo tiriamieji darbai: 2002 m. pr. Nr.1. – Klaipėda: KU leid., 2002. – P. 77–91.
5. **Eidukas D**. EP teorija // Eksperimento rezultatų įvertinimas, t.2. – Kaunas: Technologija, 2002. – P. 124.
6. **Juraitis M**., **Rudokas K**., **Toleikis Š**. Projektas „Insomnia”. Laboratorinio stendo kompiuterizuotos signalų analizės programinė įranga. – Kaunas: KTU, 2000, http://www.soften.ktu.lt/~pik98/2000-2001/insomnia/.
7. **Pham D**. **T**., **Dimov S**. **S**., **Pham P**. **T**. **N**. Research at the MEC // Robotica – UK, Cambridge Univ. Press, 2002, volume 20. - P. 563–568.

8. **Ramašauskas O**. Valdomų objektų judesių dinamikos grafinio vaizdo tyrimas // Daktaro disertacijos santrauka, Kaunas, KTU, 2002. – Klaipėda: KU leidykla, 2002. – P. 28.
9. **Ramašauskas O**., **Bielskis A**. **A**. E.laboratorija universitete. Duomenų gavyba modelio žinių bazėje // Technologijos, mokslo darbai Vakarų Lietuvoje, III. – Klaipėda: KU leidykla, 2002. – P. 100–108.

Programinės įrangos aprašymai WWW žiniatinklyje:
http://developer.java.sun.com/developer/infodocs/,
http://httpd.apache.org/docs-project/apache,
http://www.mysql.com/documentation/index.html,
http://www.php.net/docs.php,
http://www.phpmyadmin.net/documentation/phpMyAdmin.

Pateikta spaudai 2003 02 19

O. Ramašauskas, A. Bielskis. Objektų nuotolinio valdymo e.laboratorijoje aspektai // Elektronika ir elektrotechnika. - Kaunas: Technologija, 2003. – Nr. 4(46). - P. 93-99.

Plėtojant nuotolinio mokymo sistemos posistemį – e.laboratoriją, labai svarbu sukurti nuotolinio laboratorinių darbų atlikimo techninę ir programinę įrangą. Tai gana sudėtingas uždavinys, reikalaujantis daug darbo ir žinių, taikant informacinių technologijų pasiekimus mokymo procese. Šiame straipsnyje yra aprašytas užbaigtas ir praktiškai įgyvendintas projektas, valdant fizinį objektą arba jo modelį realiu laiku tipine interneto naršykle. Straipsnyje aprašyti objektai ir įgyvendinimo technologijos yra pirmasis bandymas taikyti nuotolinį valdymą Klaipėdos universiteto Informatikos katedroje mokymuisi. Tai galėtų būti bandymas pasiūlyti naują mokymosi būdą – atlikti nesudėtingus laboratorinius darbus ne laboratorijoje, o nuotoliniu būdu internete. Šis projektas padėtų kurti e.patarėją, nuotolinio mokymosi metu padedantį studentui savarankiškai papildyti savo žinias ir pasitikrinti turimų žinių lygį, interaktyviai bendraujant su šiuo e.patarėju. Toks patarėjas jau yra plačiai diskutuojamas ir tobulinamas ruošiant e.mokomąją medžiagą specialybės dalykų studijoms Klaipėdos universitete ir Vakarų Lietuvos verslo kolegijoje. Il. 8, bibl. 9 (lietuvių kalba; santraukos lietuvių, anglų, rusų k.).

O. Ramašauskas, A. A. Bielskis. Research the Remote Controlled Objects in the E.laboratory Environment // Electronics and Electrical Engineering. – Kaunas: Technologija, 2003. –No. 4(46). – P. 93-99.

The attempt to implement an adaptive learning possibility in a distance learning environment by implementing automatic distance control of a specially constructed laboratory object is being discussed in this paper. E.learning concept is being discussed for a long period in the department of Computer science of Klaipėda University. Here the term E.laboratory is being understood as a part of e.university – an environment to run distance studies. The purpose of this project was to offer a tool, which might help to start for student some experiments online. To implement such a possibility the distance interface to physical and virtual experiments in an e.laboratory must be created. It was decided to use free software for the implementation of this project. To do that, the following free software has been used in the project: Apache HTTP server, Java VM, MySQL DBCC, PHP Hypertext preprocessor and phpMyAdmin script. The TCP protocol was used to interact between two net elements – IP address and socket. To realize agent technology in communication by usage of sockets, the Java object oriented language has been used. The idea of creating a real object with a possibility to control it via Internet is discussed and the implementation of such an object in the University of Klaipėda is presented. This paper is an attempt to implement adaptive learning and modeling possibilities in a distance learning environment. The main problem appears, when the idea of flexible learning environment originates. Flexible learning environment means the system able not only to present and manage learning material, collect information about the users, give statistics on them and learning material, but even able to adapt its behavior to the specific learner needs that means it is able to analyze the learner. Specific needs are interpreted as individuality of each person, who uses the learning environment. So, it is very important to build learner's model correct as much as possible and later modify it during the changing stages of learning and interactions with virtual learning environment process. Usually standard learning environments have nothing to offer in such a case. The paper discusses the possibilities to use such an object in distance education and to add this experience to the knowledge base of e.learning adviser by studying informatics in Klaipėda University and Business College of Western Lithuania. Ill. 8, bibl. 9 (in Lithuanian; summaries in Lithuanian, English, Russian).

О Рамашаускас, А. А. Бельскис. Аспекты дистанционного управления объектами в среде е-лаборатории // Электроника и электротехника. - Каунас: Технология, 2003. - № 4(46). – С. 93-99.

В данной статье представлены определения, алгоритмы и частичные решения задач подсистемы дистанционного моделирования (составляющей системы дистанционного обучения) в университетской е-лаборатории. Определено, что в этом случае особенно важно создать и задействовать большое количество программных агентов, апплетов и сервлетов, дистанционно управляемую оболочку и клиентские интерфейсы. Показано, что педложенное проектное решение и созданное программное обеспечение позволяет решать задачи дистанционного управления физической или виртуальной моделью (лабораторным стендом) через стандартный броузер типа Internet Explorer, Netscape Navigator и др. Описанные в статье объекты и технологии - это первое приближение применения принципов управления моделями для дистанционного обучения в Клайпедском университете. Это позволяет проводить удаленный эксперимент из любого рабочего места в классе через Internet. Данный проект помогает созданию интеллектуального е-справочника для обучающихся на кафедре информатики Клайпедского университета. Илл. 8, библ. 9 (на литовском яз.; рефераты на литовском, английском и русском яз.).