

Valdomųjų objektų agentų intelektualizavimas

A. A. Bielskis, O. Ramašauskas, G. Stabingis

Informatikos katedra, Klaipėdos universitetas,

H. Manto g. 84, Klaipėda, tel. +370 46 398820, el. p. bielskis@ik.ku.lt

Ivadas

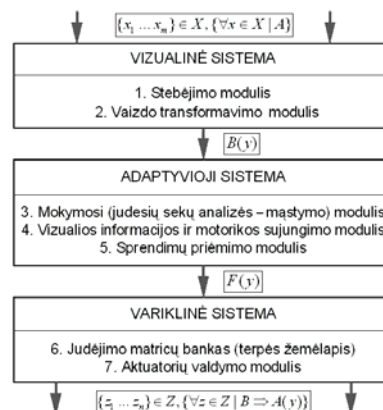
Šiuo metu sparčiai populiarėja intelektualiosios sistemos, kurios padeda valdyti įrenginius. Dažnai yra taikomos agentinės sistemos, kurios tam tikrus darbus atlieka be žmogaus įsikišimo, savarankiškai priima sprendimus, imituoja žmogaus ar gyvūno elgesį. Įvairiose srityse yra taikomas intelektualusis mokymasis, kai veikiantys programiniai agentai nustato geriausius tam tikrų darbų atlikimo variantus. Tai pritaikoma ir e. mokymuisi, kai kiekvienam vartotojui parenkamas jam tinkamiausias mokymosi metodas, ir roboto valdymui, kai apskaičiuojamas optimalus veikimo modelis: kaip išvengti kliūties, kaip tiksliau ir greičiau pernešti tam tikrą daiktą iš vienos vietos į kitą ir pan. Vienas iš Klaipėdos universitete kuriamos eksperimentinės e. laboratorijos [1] tikslų yra mokymo algoritmų planavimas, optimizavimas [2] ir įdiegimas į roboto valdymo uždavinius, pasirinkta kryptimi tobulinant e. laboratorijos struktūrą, veikimo principus ir technologijas, pritaikant sukauptą patirtį kvalifikuotiems elektronikos ir informatikos specialistams rengti.

Valdomo objekto agentinė architektūra

Judriojo roboto architektūra gali būti tobulinama taikant labai supaprastintą gyvūnų judėjimo modelį. Tokio modelio funkcijos gali būti abstrakčiai komponuojamos dirbtinių neuroninių tinklų trisluoksnyje architektūroje, remiantis pakankamai gerai ištyrinėta gyvūnų elgsenos septynių lygių neurofiziologija [3]. Pažymėtina, kad abstraktus aukšto lygio vizualios informacijos aktualizavimas galimas ir priverstinio mokymosi, intelektualiosiose skaitmeninio vaizdo sistemose (iSVS), sujungtose su programiniais arba mikroprocesoriaus valdomais vykdykliais (aktuatoriais, 1 pav.). Tokių sistemų dirbtinio intelekto modulis su ekspertinėmis naujų galimybių generavimo funkcijomis galėtų būti patiprinamas pritaikius priverstinio mokymosi algoritmus [4].

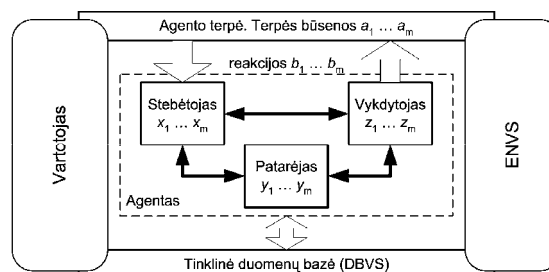
Terpės poveikių aibę galima apčiuopti jutiklių x_i signalais $\{x_1 \dots x_m\} \in X$, kuriuos atitinka poveikių verčių sąrašai $\{a_1 \dots a_m\} \in A$. Aibę X galima modeliuoti programa, pavyzdžiui, atsitiktinių skaičių generatoriumi; čia $\{\forall x \in X \mid A\}$. Analogiškai, jeigu kiekvieną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

laboratorijoje seką $A \rightarrow F(y) \rightarrow Z$ vykdančios procesai yra jos intelektualūs agentai, kurių akcijos/reakcijos priklauso nuo sudaryto valdymo algoritmo ir duomenų bazėje įrašytų žinių. Agentas – tai savarankiška, susivokianti aplinkoje programa (galinti valdyti savo veiksmus, sekdamą vieną ar daugiau objektų), gebanti mokytis (galinti pagerinti darbą, turinti adaptyvumo atributą). 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 [5, 6].



1 pav. Programiniais AI įrankiais sustiprintas iSVS modelis, imituojantis gyvūnų motorinius gebėjimus

Artinant agento „mąstymą“ prie žmogiškojo, naudotina neraiškosios logikos metodologija, susiejant jutiklių $\{x_1 \dots x_m\}$ ir programuojamųjų valdiklių $\{z_1 \dots z_m\}$ signalų apdorojimą pagal IEC 1131 rekomendacijas (2 pav.).



2 pav. E. laboratorijos agentinė struktūra tarp vartotojo ir eksperimento nuotolinio valdymo sistemos (ENVS)

Didelėse, atvirose, dinamiškose ir neprognozuojamos terpėse geriausiai veikia daugiaagentės programų sistemos su adaptyvia agentų mokymosi strategija [5]. Aprašytosios sistemos savarankiškai parenka ir įvykdo vartotojų užduotis. Agentinės programų sistemos yra autonomiškos, jos siekia savo tikslų be nurodymų ir komandų iš supančios terpės ir gali būti klasifikuojamos, atsižvelgiant į šiuos kriterijus:

- ✓ agentų skaičių (≥ 1);
- ✓ mobilumą (agentai gali judėti kompiuterių tinkle arba būti stacionarūs);
- ✓ tarpusavio sąveiką (agentai gali bendrauti tarpusavyje arba kooperuotis);
- ✓ intelektualumo lygį.

Agentai laikosi tikėjimų, troškimų ir ketinimų principo (*Belief, Desire and Intention*). Įvykus kokiam nors įvykiui, agentas į jį reaguoja remdamasis savo žiniomis (tikėjimais) ir tikslais, kuriuos nori pasiekti šiuo įvykiu (troškimais). Tikslai įgyvendinami tam tikrais, iš anksto numatytais planais (ketinimai). Vadinasi, programinis agentas gali pakeisti žmogų, nes turi galimybę išsirinkti ir priimti geriausią sprendimą. Toks agentas gali veikti savarankiškai, jis gali būti naudojamas kaip tam tikras „protingas“ asistentas, laborantas atliekant pasirinktas užduotis. Be to, agentas gali būti tam tikros sistemos valdytojas, kuris atlieka tarpininko (t.y. ENVs sąsajos dalies) tarp tos sistemos posistemų vaidmenį [3, 5]. Agentai gali būti kuriami bet kokia programavimo kalba, tačiau daugiau galimybių teikia į agentą orientuotos objektinės programavimo kalbos, specialiai pritaikytos agentams kurti [8, 9].

Objektas, funkcionuojantis dinamiškoje aplinkoje arba aplinkoje, kur galimi tam tikri pokyčiai, naujos situacijos, turi turėti mechanizmą, kuris jam padėtų prie jų prisitaikyti, sistemingai apdorotų ir sujungtų naują informaciją apie aplinką. Čia pateikiame į tikslą orientuotą agentų veikimą, kur kiekvienas agentas turi būti atsakingas už vienos ar kelių funkcijų vykdymą tuo pat metu. Funkcionalumu paremtos sistemos efektyvumą didina galimybė saugoti vidines būsenas ir mokymosi galimybė, kai kiekvienas režimo modulis gali būti pritaikytas nepriklausomai, dirbant su atskira terpės reprezentacija. Suvokimo komponentas apdoroja jutiklių duomenis ir sukuria tipinių situacijų, kylančių iš objekto aplinkos sąveikų, koncepcijas. Kohoneno neuroninis tinklas (NT) sukuria jutiklių aktyvacijos verčių blokų aibę. Blokų aibė atspindi aplinkos semantiką. Kiekvienas dirbtinis neuronas Kohoneno NT yra bloko centras ir yra aktyvus jutiklių aktyvacijų kombinacijų aibe (3 pav.). Režimo komponentas apdoroja suvokimo informaciją ir išmoka objekto valdymo, naudojant grįžtamuosius ryšius, taisykles. Jis gali susidėti iš daugybės nepriklausomų modulių, kurių kiekvienas rodo veikimo tikslą. Kiekvienas veikimo modulis yra atsakingas už tikslo pasiekimą, kuris yra aprašytas pagal jo sustiprinimo (*reinforcement*) funkciją.

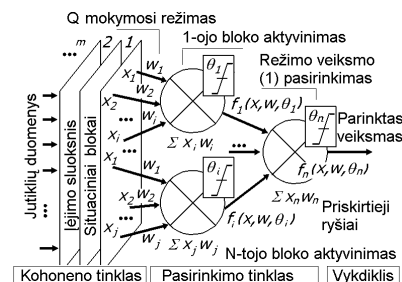
Tikslo pasiekimo procesas yra įgyvendintas veiksmo pasirinkimo ir mokymosi mechanizmu, naudojantį Q mokymosi algoritimą. Tai sustiprinto (priverstinio) mokymo algoritmas, kuris gali būti naudojamas agentams kurti ir robotams mokyti, siekiant numatyto tikslo [4, 8, 10, 11]:

$$Q(s, a) = (1 - \alpha) \cdot Q(s, a) + \alpha \cdot (r + \gamma \cdot \max_{a'} Q(s', a')); \quad (1)$$

čia $Q(s, a)$ – funkcijos Q vertė, kur s – būsena, a – veiksmas; α – mokymosi greičio koeficientas; r – paskatinimo vertė; γ – pasitikėjimo būsimomis vertėmis koeficientas; didžiausios Q vertės ateities būsena s' ir veiksmas a' – esant ateities būsenai.

Pritaikius (1) išraišką sukurtas efektyvus nesudėtingas algoritmas, pritaikytas objekto mokymosi valdymo ir optimizavimo scenarijui įgyvendinti.

1. Kiekvienai porai $s \in S$ ir $a \in A$ inicializuojamos Q vertės, įrašant nulius (vietoj nulio gali būti bet kokios atsitiktinės $Q < 1$ vertės).
2. Vykdoma iki numatyto iteracijų skaičiaus:
3. ištiriama s būsena;
4. pasirenkamas veiksmas a pagal numatytą strategiją ir įvykdomas;
5. gaunamas paskatinimas r .
6. Pereinama iš būsenos s į būseną s' ;
7. ištiriama nauja būsena s' ;
8. pasirenkamas veiksmas a' pagal numatytą strategiją ir įvykdomas;
9. atnaujinama Q lentelė pagal (1) formulę.



3 pav. Q mokymosi agentinė valdymo architektūra

Veikiantys komponentai gali būti pavaizduoti kaip nepriklausomų tikslo pasiekimo modulių, besimokančių nepriklausomai paskirstytu būdu, aibė. Q mokymosi algoritmas leidžia objektui pritaikyti veiksmo pasirinkimo strategiją kintančioje aplinkoje ir prisitaikyti, atsiradus naujai terpės situacijai. Viršutinio lygio valdymo komponentas yra sudarytas kaip pasirinkimo tinklas, įvertinantis išėjimų vertes iš sluoksnio modulių ir prižiūrintis atskirų nepriklausomų veiksmų aktyvaciją, atsižvelgiant į jų prioritetus (jautrumo slenksčius θ) apibrėžtus pasirinkimų ryšiuose. Pasirinkimo tinklas atsakingas už poveikių atrinkimą. Kai daugiau blokų tampa aktyvūs, tinklas pasirenka aukščiausio prioriteto veiksmą, kuris perduodamas objekto vykdikliams.

Kiekvienas signalas pereina pasirinkimo tinklą. Pasirenkamas veiksmas su didžiausiu prioritetu. Gautų signalų įgyto kintamojo jungiklis

$$E(t) = \sum_{i,j=0}^{N,M} \Delta \{Jutiklio_i_duomuo_j(t)\}. \quad (2)$$

Sustiprinimo signalas $r(t)$ aprašomas (3) ir (4) formulėmis pritaikytam veiksmui:

$$r_{Atmetimo}(t) = \begin{cases} -1, & \text{jei } E > 0; \\ 1, & \text{jei } E < 0; \\ 0, & \text{kitu atveju;} \end{cases} \quad (3)$$

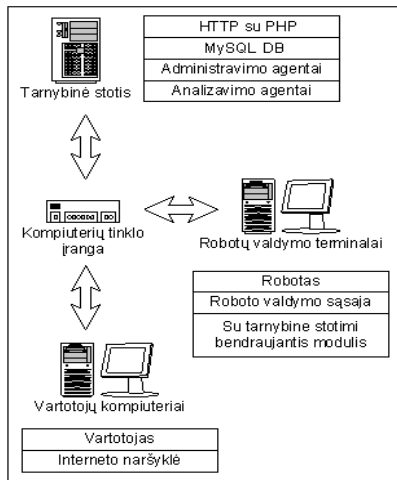
$$r_{Tvirtinimo}(t) = \begin{cases} 1, & \text{jei } E > 0; \\ -1, & \text{jei } E < 0; \\ 0, & \text{kitu atveju.} \end{cases} \quad (4)$$

Svorių pokyčiai, atspindintys pritaikytą veiksmą, apskaičiuojami ir atnaujinami pagal (1) išraiškos algoritmą. Šis ciklas gali būti baigtinis, jei sprendimo priėmimo užduotis yra epizodinė (aptikti ką nors), arba begalinis, jei užduotis yra nuolatinė (stebėti terpę). Kohoneno NT ir priverstinio mokymosi parametrai priklauso nuo objekto charakteristikų.

E. laboratorijos struktūros įgyvendinimas

Elektroninė laboratorija, kurioje mokomas agentas, sudaryta iš tokių pagrindinių komponentų (4 pav.):

- ✓ tarnybinės stoties;
- ✓ objekto valdymo terminalo;
- ✓ vartotojo kompiuterio;
- ✓ kompiuterių tinklo įrangos.



4 pav. E. laboratorijos struktūrinė schema

Tarnybinėje stotyje įdiegtas Apache programų paketas, leidžiantis vartotojams naudoti internetinę naršyklę ir palaikantis PHP scenarijus, MySQL duomenų bazių įrangą. Duomenų bazės reikalingos vartotojams registruoti, objektams (robotams) aprašyti ir kitai eksperimento informacijai saugoti [5]. Tarnybinėje stotyje veikia ir keletas programinių agentų. Vieni agentai sistemą administruoja, kiti vykdo ir analizuoja atliktus eksperimentus, nustato geriausius objekto valdymo scenarijus. Objekto valdymo terminalas – tai kompiuteris, prie kurio prijungtas eksperimentinis robotas. Šis kompiuteris turi objekto valdymo programinę sąsają, leidžiančią valdyti kompiuterio įvedimo ir išvedimo prievadus. Objekto valdymo terminalas keičiasi duomenimis su tarnybine stotimi, gauna laboratorinių darbų užsakymus ir grąžina rezultatus. Vartotojo kompiuteris – tai asmeninis kompiuteris su interneto naršykle, prijungtas prie kompiuterių tinklo, kuriame veikia e. laboratorija. Iš schemos matyti, kad objekto mokymo laboratorinius darbus galima atlikti bet kuriuo kompiuteriu, turinčiu ryšį su e. laboratorijos tarnybine stotimi. Iš struktūrinės schemos matyti, kad sistemą turėtų sudaryti mažiausiai trys

kompiuteriai. Bandomųjų darbų (programavimo, bandomojo testavimo) etape visas šias dalis galima sujungti į vieną.

Valdymo scenarijų sudarymas ir įvertinimas

Tarkime, kad tiriamo kompiuterio įtaiso, roboto įtaiso ar prietaiso automatiškai valdomą pavarą. Sudarę įtampos ir momento lygtis dvifaziam ($m = 2$) neryškių polių trumpai jungto aktyvaus rotoriaus žingsniniam varikliui, posūkio kampo atskaitą pradėkime, kai rotoriaus poliaus ašis sutampa su pirmosios fazės ašimi. Iš sinchroninės nuolatinės srovės mašinos, kurią valdytume pasirinktu algoritmu, dinaminės pusiausvyros lygčių [7] gauname stabilaus valdomo judesio išraiškas. Tiriamosios pavaros dinaminės pusiausvyros ir stabilaus judesio išraiškas bus

$$\begin{cases} r_k i_k + \sum_{j=1}^m L_{kj} \frac{di_j}{dt} + \frac{d\theta}{dt} \sum_{j=1}^{m+1} i_j \frac{dL_{kj}}{d\theta} = u_k; \\ M_a + \frac{J}{p} \frac{d^2\theta}{dt^2} = \frac{p}{2} \sum_{j,k=1}^{m+1} i_j i_k \frac{dL_{kj}}{d\theta}. \end{cases} \quad (5)$$

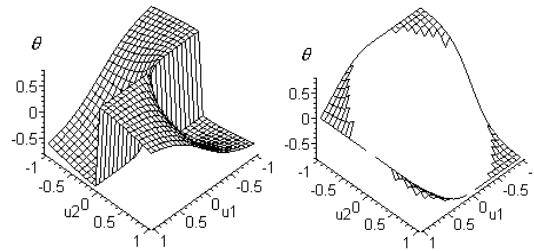
Uždavinį išsprendžiame ir vizualizuojame, naudodamiesi *Scientific Notebook* matematine programa, pasirinkę įvairias apibendrintojo koeficiento C vertes

$$u_1 \cos \theta - u_2 \sin \theta = \frac{rM_a}{p\Psi_m} = C \quad (6)$$

ir nustatę posūkio kampo θ , kurį nesunkiai galime perskaičiuoti į poslinkį, išraiškas:

$$\theta = \begin{cases} -C + \frac{u_1}{u_1^2 + u_2^2} \left(Cu_1 \pm \sqrt{u_1^2 u_2^2 - u_2^2 C^2 + u_2^4} \right); \\ \frac{1}{u_1^2 + u_2^2} \left(Cu_1 \pm \sqrt{u_1^2 u_2^2 - u_2^2 C^2 + u_2^4} \right). \end{cases} \quad (7)$$

Lygčių sprendinių laukų pavyzdžiai pateikti 5 paveikslėlyje (parodyti tik du iš n galimų).



5 pav. Išraiškų vaizdas, kai a) $C = 0$, b) $C = 1$ [1]

Reikia pasakyti, kad šio inžinerinio uždavinio tiek analitinis, tiek skaitinis sprendimas nėra vaizdas arba spartus, juolab, kad vertinant analitinio konstravimo metu atsiradusius apribojimus, nesunku suklysti. Kompiuterinių agentinio valdymo metodų taikymas daug efektyvesnis.

Aprašomoje sistemoje tiriamus objektus galima valdyti per kompiuterio lygiagrečiuosius arba nuoseklius prievadus, pritaikius arba suprojektavus originalius energijos keitimo įtaisy su mikroprocesoriais, programiniais

valdikliais. Per juos siunčiamos valdymo komandos robotui ir apdorojami gaunami grįžtamieji duomenys. Paprastai valdomasis objektas turi terpės jutiklius, poveikių dekodavimo įrangą, vykdyklius. Sprendimų priėmimo, adaptavimo ir kiti procesai atliekami terminale.

Pavyzdžiui, pavaros valdymo duomenys tokie:

- ✓ komandos pavadinimas (iki 4 simbolių);
- ✓ valdymo prievado adresas;
- ✓ į valdymo prievadą siunčiamas kodas;
- ✓ komandos aprašymas.

Duomenys grįžtamiesiems ryšiams aprašyti:

- ✓ trumpas pavadinimas (iki 4 simbolių);
- ✓ valdymo prievado adresas;
- ✓ iš valdymo prievado gaunamas kodas;
- ✓ grįžtamojo ryšio aprašymas.

Vartotojas, pasirinkęs valdomą objektą, savo kompiuterio ekrane mato veiksmo terpę, gali keisti valdymo komandas ir stebi grįžtamuosius procesus. Taip pat jis gali pasirinkti vieną iš objekto mokymo algoritmų sąrašo, o iš grįžtamųjų ryšių sudaryti rinkinį būsenų, kurias objektas turės išmokyti kontroliuoti. Vartotojas iš objekto valdymo komandų rinkinio sudaro keletą valdymo scenarijų, kurie turės keisti objekto būsenas. Scenarijus bus sėkmingas, jei jį įvykdžius, objektas pereis į planotą būseną. Sėkmingai įvykdytas scenarijus įsimenamas, tampa patikimesnis, priešingu atveju traktuojamas kaip ne toks patikimas. Objekto valdymo komandas galima vykdyti pasirinkant vykdymo intervalą, t.y. atlikti moduliaciją laiko atžvilgiu. Tai gali būti svarbu, norint nustatyti optimalų objekto veikimą. Objekto valdymo terminalas nustatyta laiką vykdo scenarijų įvertinimo eksperimentus ir atranka geriausius. Taip vartotojas galės kurti kitus, jo nuomone, patikimesnius scenarijus ir juos išbandyti.

Scenarijai vertinami pagal tam tikrus mokymo algoritmus. Vienas iš tokių yra (1) aprašytasis Q mokymosi algoritmas. Pasirinkus šį algoritmą, daugelis koeficientų generuojami automatiškai, tačiau numatyta galimybė parinkti ir įvesti svarbias charakteristikas, pavyzdžiui:

- ✓ mokymosi greičio koeficientą α ;
- ✓ pasitikėjimo būsimumis vertėmis koeficientą γ ;
- ✓ paskatinimo vertę r .

Sąsaja leidžia įvesti atsitiktinum koeficientą, rodantį, kiek procentų scenarijų bus parinkta atsitiktinai, neatsižvelgiant į jau gautus rezultatus. Taip pat galima nurodyti ir pradinę (inicializavimo) vertę Q mokymosi algoritme. Šios vertės taip pat gali būti atsitiktinės.

Vartotojas galės rinktis ir kitus mokymosi algoritmus, kurti naujus scenarijus. Vartotojų patogumui objekto valdymo sąsajos programos lange rodoma informacija apie laboratorinio darbo vyksmą. Numatyta galimybė objektą laikinai išjungti, nustatyti naudojimosi trukmę ir kt. Valdymo komandų ir grįžtamųjų ryšių duomenys, įvedami objekto aktyvavimo etape, yra pradiniai, leidžiantys kurti valdymo scenarijus ir naudotini objektui mokytis toliau.

Programiniai agentai sistemoje

Visa e. laboratorijos sistema yra agentinė sistema. Būsenas, kurias parenka vartotojai, galima interpretuoti kaip įvykius, kuriuos agentai, t.y. objekto valdymo terminalo programinė įranga, turi apdoroti. Būsenoms spręsti skirti scenarijai, kuriuos taip pat sudaro vartotojai, gali būti

apibūdinti kaip agento planai įvykiams prognozuoti. Kądangi planai pasirenkami atsižvelgiant į ankstesnius, žinomų planų vykdymo rezultatus, galima teigti, jog ši sistema yra besimokantis agentas.

Aprašytoje stadijoje agentas mokymuisi būtinas žinias gauna iš vartotojų, kurie atlieka laboratorinius darbus, kuria įvykių sekas, sudaro veiklos planus. Iškelus optimizavimo, veiklos strategijos tobulinimo, naujų tikslų generavimo uždavinius, to nepakanka. Intelektualizuojant objektų valdymą, tarnybiniuose stotyje turėtų veikti dar bent vienas programinis agentas, automatiškai kuriantis laboratorinius darbus ir vertinantis laboratorinių darbų metu gautus rezultatus taip, kaip tai darytų eilinis vartotojas. Apdorojęs pirminius duomenis, agentas imtųsi kurti tarpinius scenarijus, ieškodamas optimalaus objekto valdymo scenarijaus.

Besimokantys ir mokantys objektų valdymo agentai nėra vieninteliai agentai šioje sistemoje. Tuo pačiu metu sistemoje gali dirbti keletas vartotojų, ir valdomų objektų visiems tikriausiai neužteks. Vadinasi, vartotojai bus priversti laboratoriniams darbams atlikti reikalingus objektus užsisakyti iš anksto. Todėl reikalingas dar vienas programinis agentas – administratorius, kuris koordinuotų užsakymus ir informuotų vartotojus apie jau atliktus laboratorinius darbus. Agentas reaguotų į netikėtus pasikeitimus sistemoje. Pavyzdžiui, jeigu tiriama pavana turi autonominio maitinimo šaltinį (akumuliatorių), kurį reikia periodiškai įkrauti, tai, kol maitinimo šaltinis įkraunamas, su objektu negali būti atliekami jokie laboratoriniai darbai. Sistemos administravimo ir koordinavimo agentas privalo nukelti visus užsakytus laboratorinius darbus vėlesniam laikui ir apie tai informuoti vartotojus.

Išvados

Aprašytoji sistema leidžia nuotoliniu būdu besimokantiems asmenims pagilinti programavimo ir mechatronikos įgūdžius, atlikti laboratorinius darbus. Tyrimas parodė, kad tiek objekto valdymo terminalo, tiek vartotojo darbo vietoje pakanka minimalių išteklių, t.y. asmeninio kompiuterio, turinčio ryšį su e. laboratorijos kompiuterių tinklo tarnybine stotimi.

Nustatyta, kad analitiniais metodais spręsti objekto valdymo besikeičiančioje terpėje uždavinius gaislu, ne visada įmanoma; skaitiniai metodai spartesni, tačiau keblu išsirinkti optimalius duomenis, todėl naudojant autonomiškai veikiančius intelektualius agentus lengviau apdoroti duomenis.

Gautieji rezultatai leidžia mokomąją e. laboratoriją interpretuoti kaip didelį kompleksinį agentą, renkantį informaciją iš vartotojų ir pritaikyti ją pačiam tobulintis.

Literatūra

1. Bielskis A. A., Ramašauskas O. Taikomasis objektų modeliavimas e. laboratorijos terpėje // Lietuvos matematikos rinkinys, ISSN 0132-2818. – Vilnius: MII, 2003 – Nr. 43. – P. 185–189.
2. Eidukas D. Eksperimento planavimo teorija. – Kaunas: Technologija, 2002. – 138 p.
3. Billard A., Mataric J. M. Learning human arm movements by imitation: evaluation of a biologically inspired connectionist architecture // Robotics and Autonomous Systems. – Elsevier, 2001. – 941. – P. 1–16.

4. **Watkins C., Dyan P.** Q-learning // Machine Learning. – 1992. – Nr.8(3/4). – P. 279–292.
5. **Ramašauskas O., Bielskis A. A.** Objektų nuotolinio valdymo e. laboratorijoje aspektai // Elektronika ir elektrotechnika, ISSN 1392–1215. – Kaunas: Technologija, 2003. – Nr. 4(46). – P. 93–99.
6. **Девятков В. В.** Системы искусственного интеллекта. – Москва: МГТУ, 2001. – 352 с.
7. **Geleževičius V. ir kt.** Elektros pavarų valdymo sistemos. – Vilnius: Mokslas, 1990. – 360 p.
8. **Bielskis A. A., Stabingis G.** Objekto agentiškausiai orientuoto distancinio valdymo mokymo intelektualizavimas // Vadyba. – Klaipėda, 2004. – Nr. 1(4). – P. 9–14.
9. **The Agent Oriented Software Gr.** // JACK Documentation, <http://www.agent-software.com/shared/resources/index.html>.
10. **Dorf R. C., Bishop R. H.** Modern Control Systems. – 4th ed. – Addison–Wesley, 1998. – 832 p.
11. **Kostelnik P., Hudec M., Šamulka M.** Distributed Learning in Behaviour Based Mobile Robot Control <http://neuron-ai.tuke.sk/~kostelni/publikacie/zdroje/e-isci2002/1/node1.html>.

Pateikta spaudai 2005 05 12

A. A. Bielskis, O. Ramašauskas, G. Stabingis. Valdomųjų objektų agentų intelektualizavimas // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2005. – Nr. 8(64). – P. 55–59.

Mokymosi metu kurso laboratoriniai darbai pagilina žinias, tačiau ne visada juos įmanoma atlikti. Kartais trūksta įrangos ar tiesiog laiko. Nuotoliniu būdu studijuojantiems asmenims kurso laboratoriniai darbai yra beveik neprieinami. Siekiant išspręsti šias problemas, kuriamos elektroninės laboratorijos, kurios laboratorinius darbus leidžia atlikti internetu. Dauguma tokių e. laboratorijų yra nepakankamai adaptyvios. Klaipėdos universitete kuriama e. laboratorijos sistema taikoma valdomo objekto mokymui. Studentas galės pasirinkti robotą, surašyti jo valdymo scenarijus, o sistema, atlikusi laboratorinį darbą su objektu, pateiks gautus rezultatus. Scenarijams įvertinti bus naudojami tam tikri mokymosi algoritmai, vienas jų – Q learning algoritmas. Sistemos savarankiškumą gerintų keletas įdiegtų programinių agentų. Agentai koordinuotų ir administruotų sistemą. Taip pat agentai patys formuotų ir atliktų laboratorinius darbus, parinktų optimalius objekto valdymo scenarijus. Visą tokią sistemą galima interpretuoti kaip vieną kompleksinį besimokantį agentą, kuris padeda studentams įsisavinti žinias ir mokosi pats, siekdamas optimaliai valdyti laboratorinį objektą. Il.5, bibl.11 (lietuvių kalba; santraukos lietuvių, anglų, rusų k.).

A. A. Bielskis, O. Ramašauskas, G. Stabingis. Agent intellectualization of Controlled Objects // Electronics and Electrical Engineering. – Kaunas: Technologija, 2005. – No. 8(64). – P. 55–59.

During learning, tasks done in laboratory makes our knowledge deeper, but not always these tasks can be done. Sometimes we don't have the equipment or we are just short in time. For those, who are learning in distance, course laboratory tasks are almost impossible. To solve these problems the electronic laboratories are created, which lets to do laboratory tasks through the internet. The problem is a insufficient adaptability of the most part of e. laboratories in operation. E.laboratory system which is being created in Klaipėda University will be used for training how to control the object. Student, after choosing the object, will have to write some scripts to control the object, the system, after the laboratory task is finished with the real robot, will inform about the results. For script evaluation training algorithms will be used, one of them is Q-learning algorithm. To make the system more independent, it will have some agents. Agents will coordinate and administrate the system. Also agents will make there own laboratory tasks, to determine the most optimal object control scripts. The whole system can be interpreted as one powerful self learning agent, which helps students to learn and learns itself, to achieve his intention: to control the object as good as possible. Ill.5, bibl.11 (in Lithuanian; summaries in Lithuanian, English and Russian).

А. А. Бельскис, О. Рамашаускас, Г. Стабингис. Агентная интеллектуализация управляемых объектов // Электроника и электротехника. – Каунас: Технология, 2005. – № 8(64). – С. 55–59.

Проведение лабораторных занятий в процессе дистанционного обучения является затруднительным, но совершенно необходимым мероприятием. Часто просто невозможно выполнить работы из за нехватки оборудования или контактного времени, поэтому лицам, занимающимся дистанционно, курсовые лабораторные работы почти недоступны. С целью решения подобных проблем, создаются разные электронные лаборатории, позволяющие студентам выполнять лабораторные работы через интернет с применением стандартного броузера. В статье дано представление создаваемой э. лаборатории в Клайпедском университете, одна из задач которой намечается внедрение обучения объекта управления. После выбора объекта учащийся передаёт системе сценарии управления, а после проведения занятия система предоставляет полученные результаты. Для оценки сценариев применяются обучающиеся алгоритмы Q-learning типа. Для улучшения самостоятельности системы разработаны и внедрены управляющие и контролирующие программные агенты, задача которых состоит из формирования и проведения лабораторных работ, подборки оптимальных критериев управления и т.д. В общем предоставлении, систему можно расценивать как комплексный обучающийся агент для оптимального управления объектом. Ил.5, библи.11 (на литовском языке; рефераты на литовском, английском и русском яз.).