

ROBOTŲ TIPAI IR JŲ KARTOS

Tipai:

1. **Mobilieji**
 - 1.1. Žingsniuojantys
 - 1.2. Savaeigiai / važiuojantys
 - 1.3. Egzoskeletai
2. **Manipuliaciniai**
 - 2.1. Automatiniai
 - 2.2. Biotechniniai
 - 2.3. Interaktyvieji
3. **Kūrybiniai**
 - 3.1. Žaidžiantys
 - 3.2. Projektuojantys
 - 3.3. Kompoziciniai
4. **Informaciniai**
 - 4.1. Tyrinėjantys
 - 4.2. Atpažįstantys

Jų kartos:

1. **Pirmos kartos** robotai gali atlikti žmogaus užduotį tik jiems gerai pritaikytomis nekintamomis sąlygomis ir valdomi pagal nekintančią programą.
2. **Antros kartos** robotai turi jutiklius ir sugeba prisitaikyti prie kintančios aplinkos.
3. **Trečios kartos** robotai su dirbtiniu intelektu. Tai tobuliausia adaptyvių sistemų klasė, gebanti apdoroti plataus diapazono informaciją, kryptingiau elgtis kintančioje aplinkoje.
4. **Ketvirtos kartos** – robotai, kurie gali vystytis ir apmokytis kitus robotus, arba kurie yra sudaryti iš mechaninės ir biologinės dalių.

MULTIROBOTAI IR JŲ POREIKIS, KLASIFIKAVIMAS

Multirobotai:

1. Robotų sistema sudaryta iš daugiau nei vieno bendraujančio (bendradarbiaujančio) roboto.
2. Robotų grupė, kurioje robotai gali dalintis savo resursais, kad galėtų spręsti sudėtingas problemas.
3. Multiagentinė sistema, kurioje kiekvienas agentas yra realus fizinis robotas.

Jų poreikis:

1. Kai kurioms užduotims atlikti reikalingos robotų komandos.
 - 1.1. Robotų futbolas.
2. Kai kurias užduotis lengviau ir efektyviau spręsti pasidalinant darbą.
 - 2.1. Vietovės žemėlapių sudarymas.
3. Didesnis patikimumas.
 - 3.1. Sugedusį robotą gali pakeisti kitas komandos narys.
 - 3.2. Teritorijos išminavimui.
4. Geresnė sprendimo kokybė.
 - 4.1. Užduotį galima atlikti greičiau ir/arba tiksliau.

Klasifikavimas:

1. **Socialiai išdėstyti:** robotai galintys veikti socialinėje aplinkoje ir skirti kitus robotus nuo paprastų kliūčių.
2. **Socialiai integruoti:** robotai veikiantys socialinėje aplinkoje ir sąveikaujantys tarpusavyje ir su žmonėmis.
3. **Intelektualūs:** turi pažinimo modelį ir socialinę kompetenciją.

KOKIU TIKSLU ROBOTIKOJE NAUDOJAMAS DIRBTINIS INTELEKTAS?

Naudojant AI robotai gali apsimokyti ir atlikti sudėtingesnes užduotis, kadangi be AI robotas tiesiog tą pačią užduotį atlieka nieko nepakeisdamas, darydamas tuos pačius veiksmus.

ROBOTŲ ARCHITEKTŪROS

1. Sense-Act (SA)
2. Sense-Decide-Act (SDA)
3. Sense-Plan-Act (SPA)
4. Sense-Model-Plan-Act (SMPA)
5. Lygiagreti Sense-Plan-Act (PSPA)
6. Veiksmų pasirinkimo
7. Plano modifikavimo
8. Valdymo (Guidance)
9. Brook's architektūra
10. Sluoksninės (hibridinės architektūros)

KUO SKIRIASI TIPINIS ROBOTAS NUO INTELEKTUALAUS?

Tipinis robotas vykdydamas programą naudoja duomenis gautus iš aplinkos per įvairius jutiklius ar žmogaus per atitinkamą sąsają. Tuo pačiu jis reaguoja į aplinką ir veikia joje naudodamas įvairius vykdytuvus.

Tuo tarpu intelektualus robotas taip pat naudoja duomenis gautus iš aplinkos bei reaguoja į ją, tačiau pasitelkdamas dirbtinį intelektą jis gali kaupti žinias apie supančią aplinką ir naudotis jomis atlikdamas veiksmus.

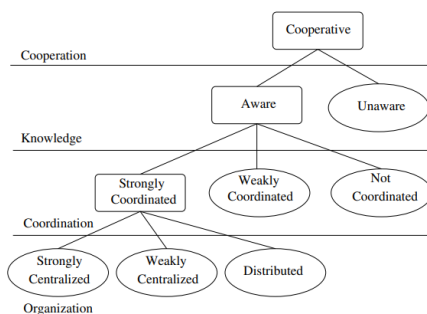
MULTIROBOTŲ STRUKTŪRA IR SISTEMOS LYGIAI

Multirobotų struktūra:

1. **Homogeninė:**
 - 1.1. Visi robotai yra tokie patys (aparatus ir valdymo programinė įranga).
 - 1.2. Esant toms pačioms sąlygoms (įėjimams) visų robotų darbo rezultatai (išėjimas) sutampa.
 - 1.3. Patikimumas: Sugedus vienam robotui, jį lengvai gali pakeisti bet kuris kitas bendruomenės narys.
 - 1.4. Adaptyvumas menkas, nes robotai nesiskiria.
 - 1.5. Naudojama spiečiaus tipo sistemose, kur laikas nėra kritinis resursas, o rezultatą galima pasiekti daug kartų kartojant tuos pačius veiksmus.
 - 1.6. Rezultato pasiekimą užtikrina tinkamas elementarių veiksmų („dėsnių“) užrašymas.
2. **Heterogeninė:**
 - 2.1. Robotų aparatinė platforma arba valdymo programa skiriasi.

Sistemos lygiai:

1. **Bendradarbiavimo** lygis.
 - 1.1. Keli robotai bendradarbiauja kartu, kad atliktų globalią užduotį, kurios negalėtų atlikti atskiras robotas, arba bendradarbiavimas leidžia pasiekti geresnes užduoties vykdymo charakteristikas.
2. **Žinojimo** lygis.
 - 2.1. Robotas žino apie kitus bendruomenės narius, apie jų veiksmus, planus.
3. **Koordinavimo** lygis.
 - 3.1. Stiprus koordinavimas – apibrėžtos koordinavimo taisyklės, kaip robotai keičiasi informacija
 - 3.2. Silpnas koordinavimas – nėra apibrėžtos koordinavimo taisyklės.
4. **Organizacinis** lygis.
 - 4.1. Stiprus centralizavimas – sprendimus visą užduoties vykdymo laiką priima tas pats lyderis.
 - 4.2. Silpnas centralizavimas - užduoties vykdymo metu lyderiu gali būti daugiau nei vienas robotas. Lyderio rolė gali būti priskiriama (keičiama) automatiškai priklausomai nuo aplinkos pokyčių arba nepatenkinamų esamo lyderio veiklos rezultatų.



ROBOTŲ VALDYMO PRINCIPAI. SPA MODELIS

SPA MODELIS

1. SPA architektūroje remiantis iš jutiklių gauta informacija suformuojamas roboto veiksmų planas.
2. Planas yra sudarytas iš komandų roboto vykdytuvams sekos.
3. „Act“ veiksmas atlieka suplanuotą veiksmų seką.

MULTIROBOTŲ ELGSENOS ŠABLONAI IR VALDYMAS

Šablonai:

1. **Susibūrimas** – pradinis etapas prieš atliekant kitas užduotis (bendrą judėjimą kartu, struktūros suformavimą).
2. **Išsiskirstymas** – tikslas yra padengti kuo didesnę teritoriją neprarandant tarpusavio ryšio.
3. **Struktūros suformavimas** – tikslas yra robotams užimti tokią padėtį erdvėje, kad jie sudarytų tvarkingą atsikartojančią struktūrą.
4. **Grandinės suformavimas** – robotai turi sudaryti grandinę, jungiančią du taškus erdvėje.
5. **Kolektyvinis judėjimas** – pagrindinė problema yra koordinuoti robotų grupę ir judėti kartu išlaikant grupės struktūrą.
 - 5.1. Robotai laikosi paprastų taisyklių:
 - 5.1.1. Vengti susidūrimo.
 - 5.1.2. Išlaikyti vienodą greitį.
 - 5.1.3. Laikytis centro.

Valdymas:

1. **Lokalus** valdymas:
 - 1.1. Paprastas, nesudėtingi skaičiavimai.
 - 1.2. Gera tinka dinaminėms aplinkoms.
 - 1.3. Ne optimalus valdymas.
 - 1.4. Labai priklauso nuo jutiklių.
2. **Globalus** valdymas:
 - 2.1. Leidžia realizuoti sudėtingesnį komandos bendradarbiavimą.
 - 2.2. Didesnė ryšio kanalo apkrova.

BROOKS ARCHITEKTŪRA

Brooks (elgsena grįsta) architektūra: sudaro keli sąveikaujančių baigtinių automatų lygmenys.

1. Brooks architektūrą sudaro užduotį įvykdančių elgsenų hierarchija
2. Kiekviena elgsena yra aprašoma paprasta taisykle
3. Kiekviena elgsena konkuruoja su kitomis elgsenomis dėl to, kad robotas ją vykdytų
4. Žemesnio lygmens elgsenos yra paprastesnės ir turi prioritetą prieš aukštesnio lygmens elgsenas

KUO SKIRIASI INDUSTRIINIAI ROBOTAI NUO BUITINIŲ. PATEIKITE PAVYZDŽIŲ

Industriniai robotai yra tikslesni, bet mažiau paslankūs ir naudojami produktų gamybai bei kol kas nenaudojamas dirbtinis intelektas.

Industriniai robotai naudojami gamybos linijų automatizavimui, mažinant žmogiškąjį faktorių, dirbant kenksmingomis aplinkos sąlygomis, o buitiniai robotai naudojami atlikti namų apyvokos darbus (kambarių valymas, baseino valymas, žolės pjovimas).

ROBOTO IR ROBOTIKOS APIBRĖŽIMAS

Robotas yra programinės įrangos kontroliuojamas mechaninis įtaisas, kuris naudoja jutiklius, norėdamas nukreipti vieną ar daugiau galinių vykdymo įtaisų, užprogramuotų atlikti judesius, kad būtų galima manipuluoti fiziniais objektais.

Robotika – mokslas apie robotų kūrimo, gamybos ir panaudojimo metodus.

IŠVARDINKITE ROBOTŲ PRIVALUMUS IR TRŪKUMUS

Privalumai:

1. Robotai gali dirbti pavojingose aplinkose.
2. Robotai dirba be jokių žmogiškų poreikių ir ligų.
3. Robotai visada turi pakartojamą tikslumą ir gali būti daug tikslesni nei žmonės.
4. Robotai ir jų jutikliai gali turėti daugiau galimybių nei žmonės.
5. Robotai gali apdoroti kelis stimulus ar užduotis vienu metu.
6. Robotai pakeičia darbuotojus, kurie gali sukelti ekonominių problemų.

Trūkumai:

1. Robotams trūksta galimybių reaguoti esant kritinėms situacijoms.
2. Robotų galimybės gali būti ribotos.
3. Robotai kainuoja brangiai.

ROBOTO ELGSENA IR JOS PLANAVIMAS

Roboto elgsena yra išorinio stebėtojo matomi roboto veiksmai (veiksmų seka)

Roboto elgsena gali būti:

1. Judėjimas.
2. Manipuliavimas su objektais.

Elgsena priklauso nuo aplinkos ir roboto būsenos.

Elgsenos planavimas:

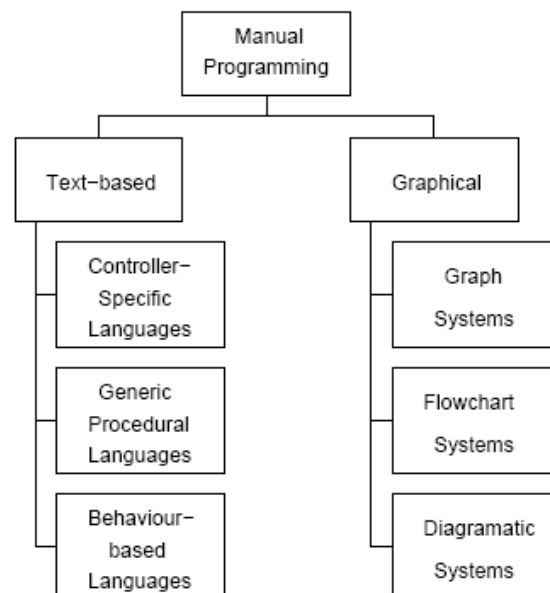
1. Analizuoti problemą.
2. Rasti bendrą sprendimą.
3. Suskaidyti sprendimą į smulkesnius žingsnius (seką).
4. Detalizuoti kiekvieną žingsnį.

APIBŪDINKITE ROBOTŲ VALDYMO PROBLEMAS

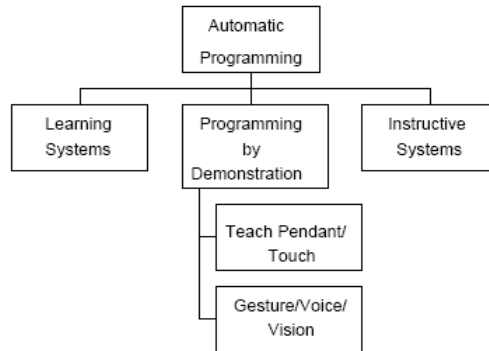
1. Sprendimų priėmimas paprastai yra lėtas, daug skaičiavimų reikalaujantis procesas.
2. Roboto reakcija turi būti greita.
3. Planavimas leidžia planuoti veiksmus iš anksto ir taip išvengti blogų sprendimų, tačiau per ilgas planavimas taip pat gali būti pavojingas (pvz., robotas gali nukristi nuo laiptų, susidurti su kliūtimi).
4. Tam kad būtų galima galvoti (samprotauti), robotas turi turėti daug tikslios informacijos apie savo aplinką, t.y. pasaulio modelį.

ROBOTŲ PROGRAMAVIMO METODAI

1. **Rankinis** programavimas
 - 1.1. Tekstinis programavimas.
 - 1.2. Grafinis (vizualus) programavimas)

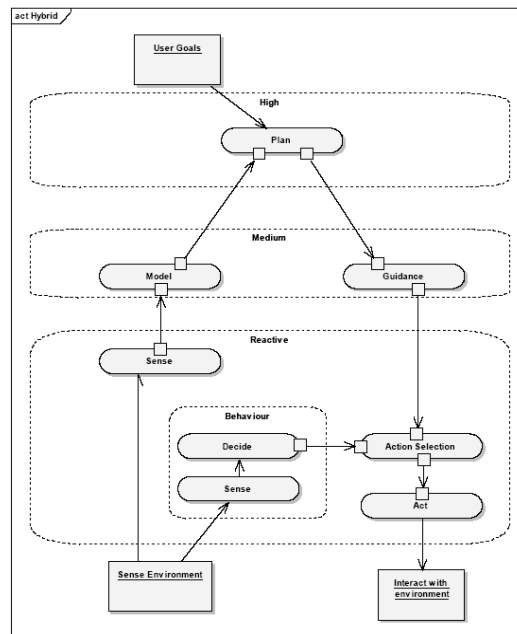


1. **Automatinis** programavimas
 - 1.1. Apsimokančios sistemos.
 - 1.2. Sistemos, programuojamos demonstruojant.
 - 1.3. Instrukcinės sistemos.



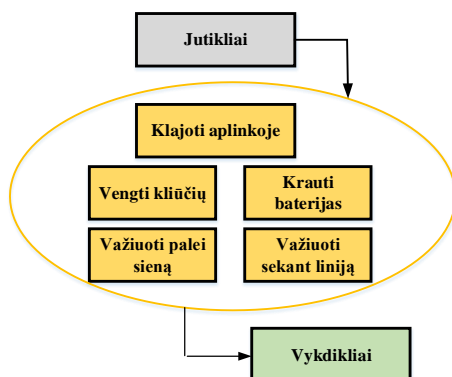
APIBŪDINKITE HIBRIDINĘ TRIJŲ LYGMENŲ ARCHITEKTŪRĄ

1. Hibridinė 3 lygmenų architektūra organizuota panašiai kaip klasikinė 3 lygmenų architektūra, tačiau žemiausias lygmuo yra pakeistas elgsena grįsta sistema.
2. Jutiklis gali aktyvuoti veiksmą elgsena grįstame lygmenyje be vidurinio arba viršutinio lygmenų įtakos.
3. Viduriniame lygmenyje gali būti reikalingas elgsenos menedžeris, kuris yra atsakingas už roboto elgsenos komponentų įjungimą/išjungimą misijos metu.
4. Jutiklio informacija naudojama modeliui sukurti, kuris savo ruožtu yra naudojamas veiksmų planui sukurti.
5. Elgsenos lygmenyje, jutimo veiksmas yra naudojamas jau sukurtam planui aktyvuoti.
6. Pastarasis gali panaikinti valdymo veiksmo išėjimus, pvz., dėl roboto saugumo.



KĄ APIBŪDINA ELLINGSSENS (TIKRIAUSIAI NORĖJO PASAKYTI „ELGSENA“) GRĮSTAS ROBOTŲ VALDYMAS?

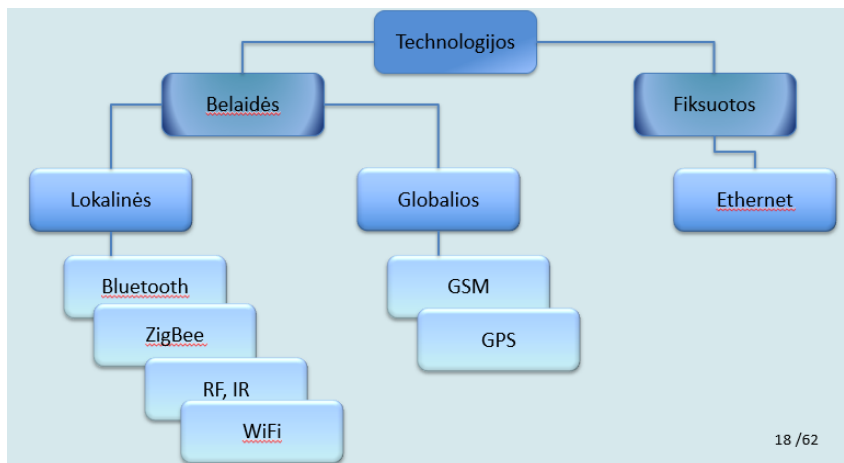
1. Robotas aprūpinimas keletu paprastųjų elgsenų, kurias gerai moka.
2. Esant tam tikroms situacijoms, jis jas kaitalioja.
3. Modelį sudaro 5 paprastosios elgsenos:
 - 3.1. Klaidžiojimas aplinkoje.
 - 3.2. Kliūčių vengimas.
 - 3.3. Važiavimas pakrauti bateriją.
 - 3.4. Važiavimas palei sieną.
 - 3.5. Važiavimas sekant išsiskiriančios spalvos liniją.



IŠ KO SUDARYTAS ROBOTŲ SENSORIŲ MAZGAS?

1. **Maitinimo blokas:** dažniausiai tai baterijos ir veikia kaip maitinimo šaltinis visiems kitiems blokams. (**nežinau ar šito reikia**)
2. **Robotų blokas:** įvairūs sensoriai informacijos registravimui.
3. **Informacijos apdorojimo blokas:** sudaro mikrovaldiklis ir atmintis. Skirta laikiniam duomenų saugojimui, kodavimui, moduliavimui.
4. **Informacijos perdavimo blokas:** skirtas užtikrinti sensorių informacijos perdavimą tarp jų ar kitų įrenginių.

APIBŪDINKITE ROBOTŲ SENSORIŲ TARPUSAVIO RYŠIO TECHNOLOGIJAS



Bluetooth - technologija skirta mažiems personaliniams tinklams. Tinkle gali būti iki 255 įrenginių, tačiau tik 7 įrenginiai vienu metu gali perduoti duomenys. Šiame tinkle **vienas** įrenginys yra pagrindinis, kiti jam pavaldūs.

ZigBee – patikimumas, mažos energijos sąnaudos, maža kaina, paprasta priežiūra, integracija su kitais standartais, saugumas, populiarumas.

IR - Infraraudonieji dažniai yra tokie, kurie randami matomo elektromagnetinio spektro gale. Jie turi ypatingai aukštą juostos plotį ir atitinkamai trumpą spindulį.

WiFi

802.11 standartas – tai bevielių lokalių tinklų technologijos standartas, nustatantis signalo perdavimo metodus fiziniame OSI lygmenyje.

ROBOTŲ KLASIFIKAVIMAS. PATEIKITE PAVYZDŽIŲ

1. **Industriniai** robotai.
 - 1.1. Automatiškai valdomas, programuojamas manipulatorius veikiantis 3 ar daugiau ašyse.
2. **Naminiai (buitiniai)** robotai.
 - 2.1. Siurblys (roomba).
3. **Medicininiai** robotai.
 - 3.1. Slaugytojai.
4. **Kariniai** robotai.
 - 4.1. Išminuotojai.
5. **Aptarnavimo** robotai.
 - 5.1. Barmenai.
6. **Laisvalaikio** robotai.
 - 6.1. Dronai.
7. **Tyrinėjimo** dronai.
 - 7.1. „Spirit“ ir „Opportunity“, tyrė Marsą.

APIBŪDINKITE ROBOTŲ ELGSENOS PLANAVIMO ETAPUS

1. Analizuoti problemą:
 - 1.1. Išsiaiškinti kokia tiksliai problemą reikia išspręsti.
2. Rasti bendrą sprendimą:
 - 2.1. Sugalvoti kaip išspręsti esamą problemą.
3. Suskaidyti sprendimą į smulkesnius žingsnius:
 - 3.1. Sugalvotą sprendimą išskaidyti į seką.
4. Detalizuoti kiekvieną žingsnį:
 - 4.1. Kiekvieną seką patikslinti.

APIBŪDINKITE ROBOTŲ VALDYMO PRINCIPUS

Robotų valdymo principai:

1. **Reaktyvusis** valdymas – negalvok, bet veik.
 - 1.1. Veikia principu – poveikis → atsakas.
 - 1.2. Tikslas – laiku reaguoti į poveikį dinaminėje ir nestruktūrizuotoje aplinkoje.
 - 1.3. Realizuojama kaip paprastas taisyklių rinkinys.

2. **Elgsena grįstas** valdymas – galvok, kaip veikti.
3. **Patariamasis** valdymas – daug galvok, o po to veik.
 - 3.1. Pagrįstas SPA modeliu.
 - 3.1.1. Nuoseklios
 - 3.1.2. Planavimas
 - 3.1.3. Atsako paieška
 - 3.1.4. Reikalingas pasaulio modelis (vidinė būseną)
 - 3.2. Trūkumai.
 - 3.2.1. Pasaulio modelis pasensta
 - 3.2.2. Planavimas ir paieška trunka ilgai
4. **Hibridinis** valdymas – galvok ir veik lygiagrečiai.
 - 4.1. Apjungia reaktyvias ir patariamąsias sistemas
 - 4.1.1. Žemutiniame lygmenyje – reaktyvioji
 - 4.1.2. Viduriniame lygmenyje – ryšio lygmuo
 - 4.1.3. Viršutiniame lygmenyje – patariamoji
 - 4.2. Lygmenys veikia lygiagrečiai
 - 4.3. Pagrindinė problema – kaip valdyti ryšius tarp lygių.

KOKIUS ŽINOTE KLASIKINIUS ROBOTŲ VALDYMO ALGORITMUS. SIENOS SEKIMO ALGORITMAS.

1. Roboto užduotis yra sekti objekto kontūrą.
2. Jei robotas aptinka savo judėjimo kelyje objektą, jis gali keliauti aplink objektą judėdamas jo perimetru.

KOKIUS ŽINOTE KLASIKINIUS ROBOTŲ VALDYMO ALGORITMUS. LINIJOS SEKIMO ALGORITMAS.

1. Linijos sekimas naudingas, kai robotui reikia pateikti iš vieno taško į kitą be jokių kliūčių savo kelyje.
 - 1.1. Galima išvengti sudėtingo kliūčių apėjimo algoritmo programavimo.
 - 1.2. Robotas gali sekti ant grindų nupieštą liniją.

KOKIUS ŽINOTE KLASIKINIUS ROBOTŲ VALDYMO ALGORITMUS. KLIŪČIŲ APĖJIMO ALGORITMAS.

1. Roboto užduotis yra išvengti kliūties.
2. Jei robotas aptinka savo judėjimo kelyje objektą, jis išvengia to objekto.

KUO SKIRIASI HORIZONTALI IR VERTIKALI ARCHITEKTŪRA.

Horizontali architektūra

Jutimo modulis interpretuoja jutiklių duomenis ir sukuria pasaulio modelį, kurį perduoda samprotavimo moduliui. Samprotavimo modulis turi tikslus ir apjungus modelį ir tikslus gaunamas planas (veiksmų seka), kuris vykstant keičia pasaulį pagal roboto tikslus.

Vertikali architektūra

Kiekvienas valdiklis mato tik tai, kas yra žemiau jo ir žino tik tai, ko jam reikia. Panašus į biologines sistemas.

KAM REIKALINGI ROBOTŲ SPIEČIAI.

1. Robotų spiečius – multirobotų sistema sudaryta iš didelio skaičiaus paprastų robotų
2. Sąveikaujant robotams tarpusavyje ir su aplinka atsiranda pageidaujama kolektyvinė sistemos elgsena
3. Paprastos elgsenos taisyklės apibrėžia sudėtingą kolektyvinę elgseną
4. Pagr. reikalavimas – nuolatinis komunikavimas ir grįžtamasis ryšys
5. Pagr. faktoriai: pigumas, paprastumas ir mažas dydis

Pageidaujamos spiečių robotų savybės:

1. Atsparumas atskiros robotų gedimui
2. Lankstumas keičiant robotų roles arba sprendžiant skirtingo tipo uždavinius
3. Priderinamumas prie įvairaus spiečiaus dydžio

KAS YRA ROBOTO BŪSENOS IR KAM NAUDOJAMAS GRĮŽTAMASIS RYŠYS

Roboto būseną yra vidinės ir išorinės būsenos kompozicija.

Vidinė būseną gali būti:

1. **Stebima:** robotas visada žino savo būseną.
2. **Nestebima/neprieinama:** robotas nežino savo būsenos.
3. **Dalinai stebima:** robotas žino dalį informacijos apie savo būseną.

Išorinė būseną: pasaulio būseną nustatoma naudojant robotų jutiklius.

Roboto saugomas būsenos vaizdas yra vadinamas vidiniu (arba pasaulio) modeliu.

Grįžtamasis ryšys naudojamas atlikti pokyčius stebint jutiklius informaciją.

1. Grįžtamasis ryšys:
 - 1.1. Teigiamas
 - 1.2. Neigiamas
2. Neigiamas grįžtamasis ryšys
 - 2.1. Stabilizuoja sistemos būseną, išieties reikšmę
 - 2.2. Jei per daug, sumažina
 - 2.3. Jei per mažai, padidina
3. Teigiamas grįžtamasis ryšys
 - 3.1. Padidina sistemos būsenos / išieties signalo reikšmę
 - 3.2. Kuo didesnė reikšmė, tuo labiau padidina

KUO SKIRIASI ROBOTŲ APMOKYMAS DEMONSTRUOJANT IR IMITUOJANT?

1. Demonstravimas
 - 1.1. Nuotolinis valdymas.
 - 1.1.1. Mokytojas tiesiogiai valdo robotą, o jo davikliai tik įrašinėja informaciją.
 - 1.2. Šešėliavimas.
 - 1.2.1. Sistema bando atkartoti mokytojo judesius stebėdama savo davikliais.
2. Imitavimas
 - 2.1.1. Davikliai ant mokytojo kūno įrašinėja mokytojo judesius.
 - 2.1.2. Išoriniai davikliai įrašinėja mokytojo judesius.

APIBŪDINKITE BREITENBERGO MAŠINĄ

Paprastos konstrukcijos roboto modelis, galintis pademonstruoti sudėtingą elgseną

1. Mobilus robotas, simetriškas bent vienos ašies atžvilgiu.
2. Turi du ratus.
3. Turi bent du jutiklius.
4. Paprasta reaktyvinio tipo elgsena.

Analogas gyvojoje gamtoje: E. coli bakterija.

Breitenbergo mašina paprastų elgsenų pagalba leidžia modeliuoti robotų „emocija“. Breitenbergo mašina įrodo, kad net ir paprasto architektūros robotas gali turėti sudėtingą elgseną, panašią į primityvių biologinių būtybių elgseną.

KOKIU TIKSLU NAUDOJAMAS ROBOTŲ MODELIAVIMAS?

Sprendžiant šias problemas:

1. Techninė robotų dalis yra brangi
2. Aparatūrą sunku testuoti
3. Įvairūs apribojimai gali neleisti dirbti lygiagrečiai

Pranašumai:

1. Lengva išmokti.
2. Lengvas prototipų kūrimas.
3. Geras mokymosi ir tyrimų įrankis.

Trūkumai:

1. Nėra „triukšmo“ – idealizuotas pasaulio vaizdas
2. Nepilni arba netikslūs duomenys
3. Priderinimas reikalauja daug laiko

APIBŪDINKITE KELETĄ ROBOTŲ PROGRAMAVIMO APLINKŲ IR KALBŲ

Microsoft Robotics Developer Studio (C#)

Windows platformai sukurta aplinka, skirta robotų kūrimui. Ji yra integruota su Visual Studio .NET, todėl programuoti galima bet kuria .NET palaikoma programavimo kalba.

Pranašumai:

1. Palaikomi paskirstyti skaičiavimai.
2. Nepriklauso nuo robotų techninės įrangos, lustų architektūros.
3. Galima valdyti sudėtingus robotus turinčius daugiau nei vieną procesorių.

V-rep

Bendrosios paskirties robotų imitatorius su integruota kūrimo aplinka. Gali modeliuoti ir imituoti jutiklius, mechanizmus ir robotines sistemas.

Taikymai:

1. Greitas prototipų kūrimas
2. Greitas algoritmų kūrimas
3. Sistemų imitavimas
4. Demonstravimas

mBlock

Grafinė programavimo aplinka sukurta Scratch 2.0 pagrindu.

1. Paprasta, intuityvi Scratch 2.0 grafinė sąsaja.
2. Bevielis ryšys: Bluetooth arba 2.4GHz bevielis ryšys komunikavimui su robotu
3. **Arduino režimas:** galimybė lengvai pereiti į programavimą tekstiniu režimu
4. Lengva naudoti.