Riadenie robota rozpoznávaním ľudských pohybov pomocou zariadenia Kinect

[1]Peter ADAMONDY, [2]Peter PAPCUN, [3]Ján JADLOVSKÝ

[1,2,3]Katedra kybernetiky a umelej inteligencie, FEI TU v Košiciach, Slovenská republika

[1]peter.adamondy@student.tuke.sk, [2]peter.papcun@tuke.sk, [3]jan.jadlovsky@tuke.sk

***Abstrakt* — Predkladaná práca pojednáva o návrhu a vytvorení modulárnej aplikácie riadenia robota rozpoznávaním ľudských pohybov pomocou senzora kinect. Riadiť budeme robotickú ruku RV-2SDB od spoločnosti Mitsubishi, ktorá je v súčasnosti integrovaná v pružnej výrobnej linke. Navrhneme nový prístup k vytváraniu pracovných sekvencií, pričom na ovládanie použijeme ruku operátora, ktorej pohyb bude zaznamenaný a použitý ako predloha trajektórie robotického ramena.**

***Kľúčové slová* — robot, riadenie, kinect, rozpoznávanie ľudských pohybov.**

I. ÚVOD

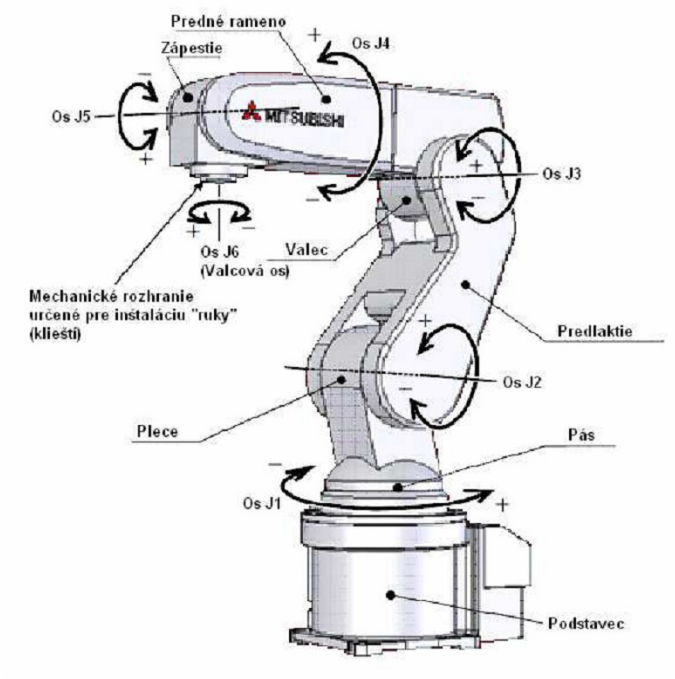
Je prirodzené, pri riešení akejkoľvek úlohy využiť najmodernejšie postupy a technológie. Ak sa pozrieme na historicky vývoj, ktorým si svet technológii prešiel, nemôžu nám uniknúť zmeny v prístupoch človeka k nástrojom vlastnej prace. Je v ľudskej prirodzenosti vytvárať nástroje, ktoré uľahčujú a zrýchľujú jeho činnosť. Najmä v moderných ergatických systémoch si však často kladieme otázky súvisiace s interakciou medzi človekom a strojom. Preto vytvoríme systém riadenia robota, tak aby bol schopný kopírovať pohyby užívateľa. Aplikácia umožni tieto pohyby uložiť a použiť ich pre programovanie rutinných sekvencií, ktoré robot ma vykonávať. Takýto prístup dovolí extrémne rýchle preprogramovanie robota na novu úlohu pri minimálnom úsilí. Tiež je možne takéto riadenie použiť na operácie, ktoré sa veľmi zložito programujú, alebo priamo vyžadujú účasť obsluhy na vykonávanej úlohe, ktorá však nesmie, či nemôže fyzicky vkročiť do miesta samotného prevádzania operácie

II. ROBOT MITSUBISHI RV-2SDB

Mitsubishi RV-2SDB je kompaktný, modulárny robot využívaný v priemyselných výrobných linkách na rôzne operácie (presúvanie materiálu, rezanie, lakovanie atď.). Ma šesť stupňov voľnosti, s maximálnou váhou bremena 3kg. Fotografia robota je na Obr. 1. Nachádza sa v laboratóriu V147 Katedry kybernetiky a umelej inteligencie, kde je súčasťou pružnej výrobnej linky. Viac v [3].

Obr. 1: Robotické rameno Mitsubishi RV-2SDB

Riadenie zabezpečuje riadiaca jednotka Mitsubishi CR1D, s ktorou PC komunikuje pomocou Ethernet pripojenia. Samozrejme, riadiaca jednotka dokáže pracovať aj s inými sieťovými rozhraniami (ProfiBus, RS-232)

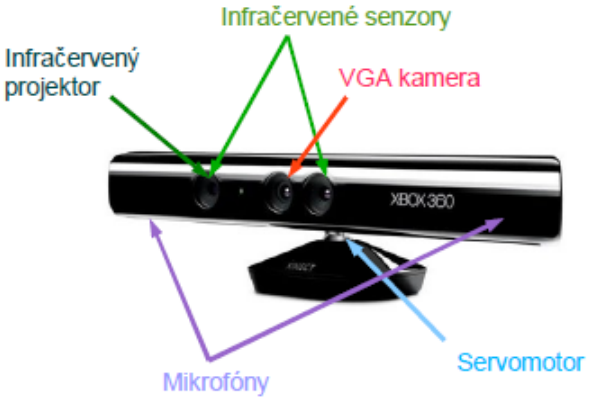


Obr. 2: Osi robota Mitsubishi melfa RV-2SDB.

III. KINECT

Kinect je špeciálne vstupne zariadenie pôvodne určené pre hernú konzolu Microsoft Xbox 360. Jeho úlohou je snímať pohyby tela hráča a prenášať ich do herného prostredia. Na trh sa Kinect dostal v novembri 2010. Vzhľadom k veľkému záujmu využitia tejto technológie aj k iným účelom než k hraniu hier, bol v júni 2011 vypustený Kinect SDK (Software Developer Kit). Jednalo sa o balíček programátorských nástrojov slúžiacich na integráciu kinectu do užívateľských aplikácii.

Aj keď bol vyvinutý spoločnosťou Microsoft, jeho najdôležitejšia časť pochádza od izraelskej firmy PrimeSense, ktorá vyvinula technológiu ovládania elektronických zariadení pomocou infračerveného projektora a snímača. Túto technológiu nazvala LightCoding.



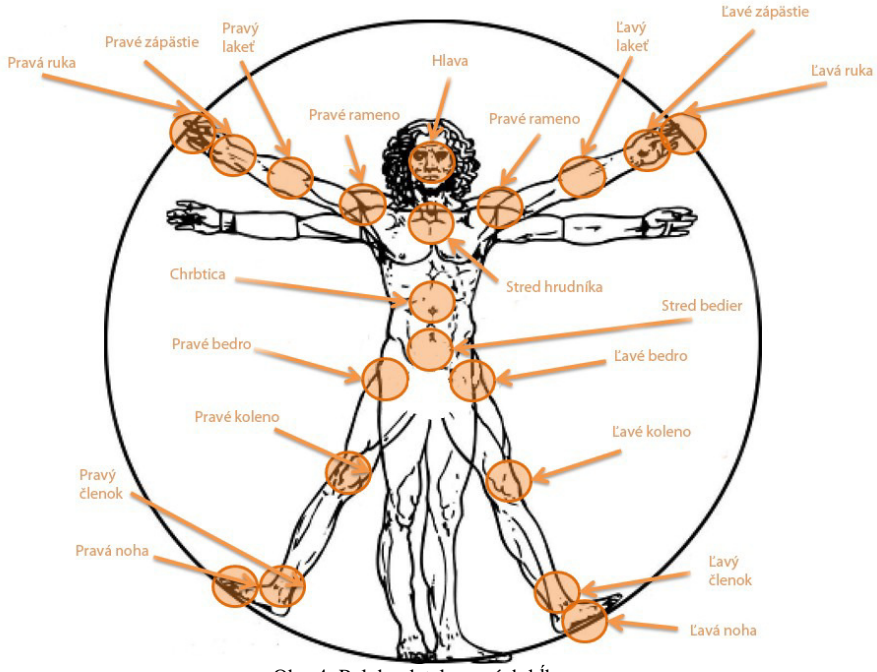
Obr. 3: Rozloženie jednotlivých prvkov zariadenia Kinect

Projektor vysiela do priestoru infračervené lúče. Senzor potom zachytáva ich odraz od povrchu. Intenzita odrazeného infračerveného žiarenia v kombinácii so vzájomnou vzdialenosťou týchto bodov vytvára hĺbkovú mapu.

IV. ROZPOZNÁVANIE KOSTRY

Rozpoznávanie kostry využitím hĺbkovej mapy prináša niekoľko výhod oproti tradičným technikám rozpoznávania z farebných obrázkov. Môžu pracovať pri akýchkoľvek svetelných podmienkach, bez ohľadu na farbu či textúru. Zjednodušujú pracovanie obrazu a tým aj odstraňovanie pozadia. Najvýznamnejšou výhodou je jednoduchá extrakcia hĺbkovej siluety užívateľa, čo umožňuje ľahké získavanie trénovacích dát. Algoritmus detekcie kostry a pohybu bol navrhnutý týmom na Cambridgeskej univerzite v spolupráci so spoločnosťou Microsoft [1]. Syntetické dáta získané z algoritmu používajúcom motion capture dáta, vytvára hĺbkovú mapu renderovaním 3D modelov postav. Takto získané dáta slúžia na trénovanie rozhodovacích stromov zlúčených do rozhodovacieho lesa. Pred tým než sa však dajú použiť, musia sa vymedziť časti tela užívateľa. Ide o definovanie zón na povrchu siluety postavy. Niektoré z týchto zón predstavujú oblasti v ktorých sa nachádzajú kĺby, iné vypĺňajú priestor medzi nimi. Silueta je v algoritme rozdelená na 31 oblastí, z toho 20 oblastí predstavuje možnú polohu kĺbu. Iba kvôli presnosti je nutne podotknúť, že pojem „*kĺb*“ predstavuje všetky body ohybu kostry a teda nie len fyzické kĺby užívateľa.

Výsledkom algoritmu sú súradnice dvadsiatich kĺbov v 3D priestore. Tieto kĺby, resp. „body ohybu“ sú znázornené na Obr. 4.



Obr. 4: Polohy detekovaných kĺbov

V. NÁVRH ARCHITEKTÚRY APLIKÁCIE

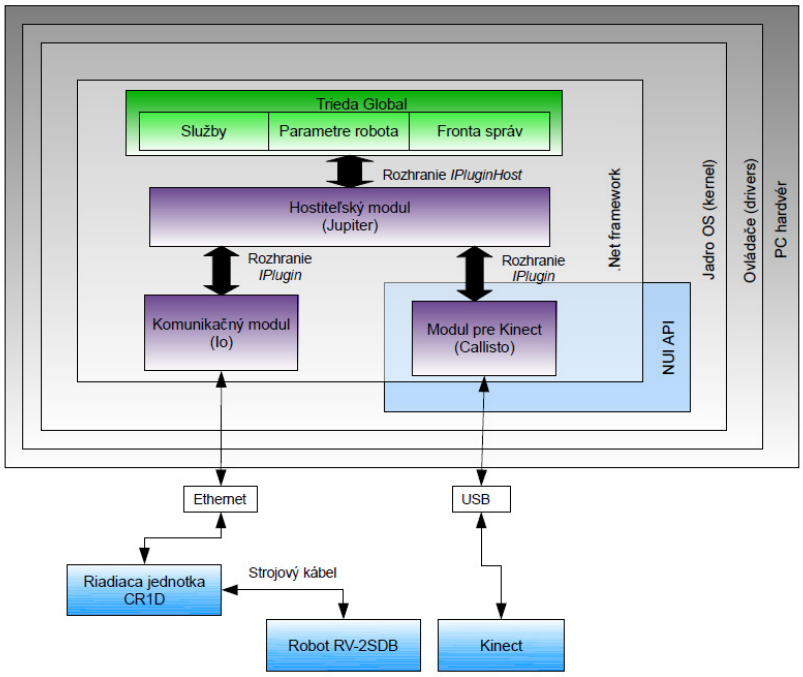
Ak chceme vytvoriť moderný riadiaci systém, musíme vytýčiť základné piliere dizajnu, resp. požiadavky na funkcionalitu. Keďže reálne využitie radenia robotov pomocou rozpoznávania pohybov ešte ani nevieme plne predvídať (vytváranie automatizovaných sekvencií, operácie pacientov na diaľku, presun a manipulácia materiálu v neprístupných miestach atď.), musí byť systém dostatočne univerzálny, aby sa vedel prispôsobiť čo najširšiemu spektru úloh. Je tu teda požiadavka modularity riešenia.

V praxi dochádza často k situácii kedy je nutné meniť systémy riadenia počas samotnej prevádzky. Možnosť pridávať, či upraviť funkcionalitu systému, bez nutnosti zastavenia prevádzky, môže nie len uľahčiť implementáciu nových častí, ale aj ušetriť náklady spojené s odstávkou celého systému. Definujme teda potrebu meniť vlastnosti aplikácie za behu, ako ďalšiu požiadavku.

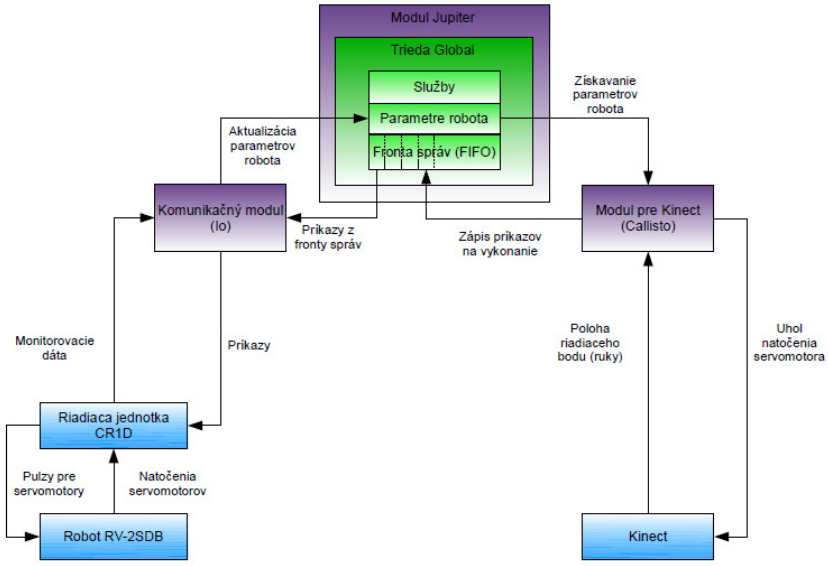
Z hore spomenutých podmienok dizajnu vyplýva potreba použiť princípy modulárneho programovania. Modulárne programovanie nie je žiadnou novinkou. Jeho implementáciu nájdeme v mnohých moderných informačných systémoch, aplikáciách, či aplikačných serveroch. Modulárny systém je taký, ktorého funkcie sú tak povediac zabalené do modulov (V angličtine sa dajú nájsť aj pod pojmom „plug-ins“, preto sa v aplikácii odkazujem na moduly ako na pluginy). Každý z týchto modulov má v rámci systému svoju úlohu a skladá sa z dvoch logických častí. Prvou časťou je samotný vnútorný systém modulu, ktorý sa môže od ostatných

líšiť *de facto* ľubovoľne. Druhá časť je vonkajší interfejs. Teda rozhranie ktorým modul komunikuje s vonkajškom. Interfejs funguje ako „lepidlo“ medzi modulmi.

Pre realizáciu návrhu architektúry celého systému sme museli vyriešiť, akým spôsobom bude nami vytvorená aplikácia komunikovať s hardvérovými prvkami. Aplikácia využíva .Net framework, ako skupinu štandardných knižníc nad operačným systémom. Modul Callisto okrem toho k získaniu dát implementuje aj niektoré prvky NUI API (Natural User Interfaces). Vďaka nemu môžeme pristupovať na senzor kinect. Globálna schéma systému vrátane komunikačných rozhraní je na Obr. 5.



Obr. 5: Globálna architektúra systému



Obr. 6: Komunikačný dátový model systému

Model na Obr. 6 reprezentuje kolobeh dát v rámci systému. Ako vidíme, sú tu dva uzavreté obvody riadenia. Prvá spätnoväzobná slučka (vnútorný obvod) sa skladá z regulátorov v riadiacej jednotke CR1D a servomotorov v robotovi. Druhý (vonkajší) riadiaci obvod sa skladá z modulu Callisto a riadiacej jednotky, ako riadeného systému. Vonkajší riadiaci obvod generuje trajektóriu (ako sled príkazov posielaných modulom Io) pre riadiacu jednotku robota. Treba poznamenať, že vnútorná slučka je z vonku neprístupná, t.j. nemôžeme ju nijako meniť (bola vytvorená spoločnosťou Mitsubishi).

Treba si uvedomiť, že sa tu stretávame s dvoma typmi modulov. Jeden predstavuje hostiteľa a druhý je hosťom. Zatiaľ čo hostiteľ je vždy len jeden, hostí môže byť viac. Hostia využívajú služby hostiteľa a on ich informuje o rôznych udalostiach. Pre jasnejšie vysvetlenie si

predstavme nasledujúci príklad. Užívateľ sa rozhodne zmeniť jazyk aplikácie. Každý modul v sebe cez interface implementuje metódu na zmenu jazyka s názvom „zmenJazyk()“. Hostiteľský modul upozorní všetky ostatné moduly tým, že túto metódu zavolá. Výhodou použitia interface [2] je práve nutnosť implementácie všetkých metód, ktoré sú v ňom definované. Hostiteľ sa teda nemusí obávať, že niektorý modul túto metódu nemá.

VI. MEDZIMODULOVÁ KOMUNIKÁCIA

Moduly sú samostatné objekty s vlastnou vnútornou logikou. Navonok sa však musia chovať uniformne. Keďže moduly majú svoje úlohy, musí byť zabezpečená vzájomná komunikácia. Všetky informácie budú prechádzajú hostiteľským modulom, avšak ho nijako nesmú zaťažovať. Môžeme použiť metaforu s hostiteľom. Najlepším riešením by bola možnosť vymeniť si informácie tak, že jeden hosť zapíše do hostiteľskej „knihy návštev“ informáciu a druhý hosť si ju vyčíta keď potrebuje. V podstate hostiteľ v tejto komunikácii hrá rolu „strážcu knihy“. V rámci systému bude dostupná fronta správ pre všetky moduly, ktorýkoľvek do nej môže zapisovať a vyberať informácie podľa potreby. Fronta je generická, t.j. správy sa môžu líšiť. V našom prípade budeme posielať príkazy z modulu riadenia (Callisto) do komunikačného modulu Io. Io bude zas zodpovedný za aktualizáciu informácií o robotovi.

VII. ZÁVER

V tejto práci sa ukázalo, akým spôsobom sa dajú využiť pohyby človeka na ovládanie robotického ramena. V budúcnosti by sa tento, alebo podobný systém, dal využiť pri operáciách pacientov. Robot môže nahradiť ruku človeka v miestach, do ktorých sa operátor, či užívateľ nemôže dostať. Ďalšie využitie vidím v rýchlom preprogramovaní robotov vo výrobných linkách. Čitateľa iste napadnú vlastné možnosti aplikácie.

Zhrňme si teda, čo sa nám v tejto práci podarilo. Analyzovali sme možnosti rozpoznávania pohybov človeka pomocou kinectu. Detekciou a následnou transformáciou polohy ruky užívateľa sme vytvoril mechanizmus riadenia v rámci modulárnej aplikácie. Návrh tejto aplikácie umožňuje počas chodu vypínať a zapínať moduly podľa potreby. Ak by sme napríklad chceli pridať riadenie nového robota, jednoducho by sme vytvorili nový modul, ktorý by komunikoval s riadiacou jednotkou, pričom by ostatné moduly zostali rovnaké. Vieme si predstaviť rozšírenie tohto systému napríklad modulom, ktorý by na ovládanie používal joystick, myšku, trackball, alebo v podstate akékoľvek iné vstupné zariadenie. Návrh takéhoto typu riadiaceho systému nesporne prináša niekoľko výhod, keďže urýchli a zjednoduší vývoj nových riadiacich a monitorovacích aplikácií, čím ušetrí čas a finančné prostriedky.

POĎAKOVANIE

Táto práca bola vytvorená realizáciou projektu Rozvoj Centra informačných a komunikačných technológií pre znalostné systémy (kód ITMS projektu: 26220120030) na základe podpory operačného programu Výskum a vývoj financovaného z Európskeho fondu regionálneho rozvoja.

REFERENCIE

[1] SHOTTON, J. – Fitzgibbon, A. et al.: Real-Time Human Pose Recognition in Parts from Single Depth Images, Microsoft Research Cambridge & Xbox Incubation, 2010. Dostupné na internete: http://research.microsoft.com/pubs/145347/BodyPartRecognition.pdf

[2] SHARP – JOHN: Microsoft Visual C# 2008. Brno : Computer Press, 2008, s.232-235. ISBN 978-80-251-2027-9

[3] PAPCUN, P.: Riadenie robota integrovaného v pružnej výrobnej linke. Diplomová práca. Košice: Technická univerzita v Košiciach, Fakulta elektrotechniky a informatiky, 2011. 95 s.

[4] ADAMONDY, P.: Riadenie robota rozpoznávaním ľudských pohybov pomocou zariadenia Kinect, Diplomová práca. Košice: Technická univerzita v Košiciach, Fakulta elektrotechniky a informatiky,2012. 76s.