

Posudek oponenta

Bakalářská práce Tomáše Chaloupeckého „Řídicí rozhraní pro roboty MITSUBISHI v prostředí ROS“

Oponent: Mgr. Martin Pecka, Ph.D. (Katedra kybernetiky, FEL ČVUT)

Návrh klasifikace: A (výborně, excellent)

Předložená bakalářská práce je kvalitním výsledkem práce studenta. Zadání splnil beze zbytku a na vysoké úrovni. Zadání bylo orientováno s výhledem na budoucí používání manipulátorů pro výuku, a zdá se pravděpodobné, že vyvinutý software a dokumentace půjdou pro výuku opravdu použít bez potřeby větších změn.

Jazykově je práce povedená – čte se dobře, text na sebe logicky i mluvnicky navazuje. V textu zůstalo několik chyb a překlepů, což je škoda, protože většinu z nich by odhalil spellchecker (jeho použití před odevzdáním práce by se mělo stát součástí standardních postupů, které student používá při psaní prací). Zvolený jazyk práce – češtinu – nevnímám jako šťastné rozhodnutí (ovšem student má formálně právo si český jazyk zvolit, proto mu nepřikládám váhu v hodnocení). Obecně je v technických oborech uznávána angličtina, a práce psané v ní mají mnohem větší šanci na to, aby si jich někdo všiml a obohatily ho. Navíc konkrétní doména této práce (ovladače pro ROS) nemá ustálenou českou terminologii, a bakalářská práce by ji určitě vymýšlet neměla. Např. studentem použitý výraz „topik“ není slovo ani české, ani anglické. Souhlasím, že „téma“ není vhodný překlad, ale „topik“ je ještě horší. V takovém případě bych se přimlouval za použití anglického termínu bez počeštění s poznámkou, že vhodný český termín neexistuje. Tohoto a mnoha dalších dilemat by byl student ušetřen, kdyby text vypracoval v angličtině. Oceňuji, že alespoň dokumentace v angličtině je – předpokládám, že to byl požadavek pro použitelnost ve výuce. Další připomínku mám k tomu, že je text psán v první osobě jednotného čísla. Pro formální technické práce je lepší používat první osobu množného čísla (ač student pracoval sám), případně trpný rod (ač to někdy může být náročné). A ještě drobná jazyková poznámka: „.urdf“ není typ souboru, ale jeho přípona. Typ je v tomto případě „URDF“ (obdobně „.xml“, „.stl“ aj.).

Text je členěn do osmi kapitol, které pokrývají zadání práce. Členění do kapitol a podkapitol pomáhá orientaci v textu. Na začátku práce je vysvětlený použitý formalismus, a ten odpovídá standardním formalismům používaným v technických a matematických textech. Následuje kapitola o middleware ROS, která vhodně a stručně uvede do problematiky čtenáře ROSu neznalé. Je ovšem jasné, že tento stručný úvod nemusí začátečníkům stačit, a proto mi v textu chybí odkaz na oficiální ROS a Gazebo tutoriály. Práce pokračuje vysvětlením výpočtů přímé a inverzní kinematické úlohy, které jsou vhodně ilustrované nákresy a obrázky pro lepší pochopení použitých konceptů. Následuje nejrozsáhlejší kapitola o implementaci ROS rozhraní. Na závěr je ještě zmíněno zjednodušené ovládací rozhraní pro studenty a s jeho pomocí navržena a ukázkově vyřešena praktická úloha.

Seznam použité literatury pokrývá text práce dostatečně. Formátování seznamu literatury je nekonzistentní, u některých anglických názvů chybí kapitalizace tématu práce. U Mitsubishi manuálů by bylo vhodné uvést i webové odkazy. Odkazy na ROS a Moveit by neměly směřovat na stránky s tutoriály, ale na obecné stránky představující dané frameworky.

K řešení student přistoupil zodpovědně a důkladně nastudoval standardy používané v ROSu pro rozhraní robotických manipulátorů. Správně se rozhodl vytvořit ovládací rozhraní pomocí balíku Moveit a rozhraní pro vykonávání příkazů pomocí `ros_control`. Tím je zajištěna maximální možná míra integrace vypracovaného kódu s dalším software. Jen je v textu zmínka, že práce byla vypracována pro ROS Melodic, což měla být v době zadání práce nejnovější verze ROSu. To ovšem není pravda, neboť v dubnu 2020 byl vydán ROS Noetic. Zrovna Melodic a Noetic se od sebe dost liší (např. verzí Pythonu, OpenCV nebo FCL knihovny), proto by chtělo použití ROSu Melodic zdůvodnit lépe. Programový kód rozhraní pro ROS je funkční a dobře členěný. Občas jsou použity nevhodné názvy (např. je implementován ROS balíček „kinematics_6DOF“, ale ROS balíčky nemají v názvu obsahovat velká písmena). Metadata balíčků jsou ve velmi špatném stavu. Všechny `CMakeLists.txt` a `package.xml` soubory jsou plné zakomentovaného balastu z tutoriálů, a student se nikde nevyplnil jako maintainer vytvořených balíčků (v některých figuruje jeho jméno, ale s emailem chatom@todo.todo). Stejně tak licence většiny balíčků je TODO, což je spíše poznámka, že se o licenci má teprve rozhodnout. Balíček „schunk_pg70_hardware_interface“ je dokonce kvůli balastu v `CMakeList.txt` nemožné zkompilovat (očekává adresář „include“, který ale součástí balíčku není). Manuální vytvoření chybějícího adresáře pak již dovolí kompletní kompilaci všech balíčků. Ovšem ne podle dokumentace k pracovnímu místu, kde je místo nejdůležitějšího příkazu „catkin build“ uvedený příkaz „catkin install“, který v daném kontextu nedělá nic užitečného. Student také na různých místech odkazuje na workspace sestavený buď v „devel“ režimu nebo v „install“ režimu – to by rozhodně chtělo sjednotit (navíc mám silné pochybnosti, že by workspace v režimu „install“ vůbec fungoval – opět kvůli nedůsledné práci s metadaty v `CMakeLists.txt`). Workspace v „devel“ režimu se mi ale podařilo rozchodit a simulace a ovládání všech robotů v něm fungovalo, jak má. Tím se zároveň prokázalo, že spočítaná řešení kinematických úloh jsou správná. Student pro roboty vytvořil kvalitní vizualizační modely. Jako kolizní modely použil zjednodušené meshe, což je sice přípustné, ale z hlediska výkonu neoptimální. Lepší by bylo kolizní model sestavit z geometrických primitiv. Při použití meshů je potřeba brát v potaz, že pro některé úlohy používá Moveit jejich konvexní obal, což může přinést nečekané chování.

Student splnil všechny body zadání. Kromě prozkoumání dokumentace robotů a standardních rozhraní pro ovládání manipulátorů i jedno takové rozhraní implementoval. V implementaci využil navrženou metodu analytického výpočtu inverzní kinematiky. Navíc vznikla ještě podrobná dokumentace k oběma pracovním místům pro budoucí studenty pracující s roboty v rámci výuky. Nad rámec zadání pak student přidal ještě ukázkovou úlohu stavění pyramidy z kostek, která prokazuje správnou implementaci kinematiky a řízení, a může sloužit i jako jedna z úloh ve výuce.

Jelikož vzniklý software může být užitečný pro univerzity po celém světě, doporučoval bych jeho zveřejnění na Githubu nebo jiné platformě pro sdílení kódu. Stejně tak bych doporučil vydat software jako oficiální ROS balíčky (to by ještě vyžadovalo drobné úpravy, s nimiž mohu poradit).

Posudek vypracoval Martin Pecka dne 30. 5. 2021 v Praze.