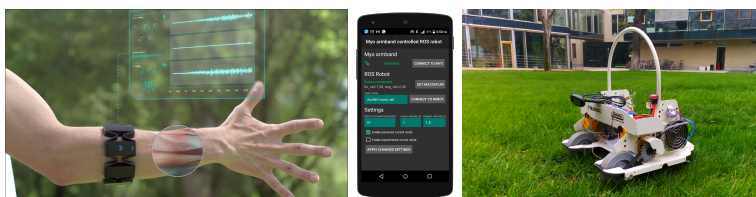


Nositelná zařízení a jejich aplikace při ovládání robota

Jan Herec*



Abstrakt

Nositelná zařízení jsou dnes aktuálním tématem. Některá mají úzkou specializaci a oblast použití, jiná jsou více obecnější a jejich potenciál je tak mnohem větší. Cílem práce je představit nositelná zařízení a jejich možné uplatnění při ovládání robota. Práce se tak blíže zabývá obecně využitelným nositelným zařízením s názvem Myo armband. Toto má tvar náramku, který se umístí na předloktí a je schopno bezdrátově pomocí technologie Bluetooth LE (Low Energy) zasílat jinému zařízení data ze svých senzorů jako jsou EMG (elektromyografické senzory pro snímání svalové aktivity), magnetometr, gyroskop a akcelerometr. Práce konkrétněji popisuje aplikaci učenou pro platformu Android, která data z náramku přijímá, vyhodnocuje a pomocí technologie Wi-fi zasílá řídicí zprávy robotovi. Aplikace je navržena pro ovládání pojízdného robota, který je schopný jezdit dopředu, dozadu a zatáčet a to na základě gest a pohybů ruky, na niž je připevněn náramek. Blíže je popsán také robot, na kterém bylo ovládání otestováno. Je potřeba zdůraznit, že je takto pomocí aplikace a náramku možné ovládat jakéhokoli pojízdného robota poháněného Robotickým operačním systémem (ROS).

Klíčová slova: Nositelná zařízení — Myo armband — ROS — Ovládání robota — Android aplikace

Příložené materiály: [Demonstration Video](#) — [Downloadable Code](#)

*xherec00@stud.fit.vutbr.cz, Faculty of Information Technology, Brno University of Technology

1. Úvod

Nositelná zařízení (v angličtině také wearable devices nebo jen wearables, češtině také nositelná elektronika) jsou elektronická zařízení, navržena tak, aby je člověk mohl nosit na svém těle. Tato zařízení mohou být ve formě doplňku, který je možné upevnit na tělo, respektive jeho část a nebo mohou být součástí oblečení. Nejedná se o přenositelná zařízení (možno nosit v kapse, držet v ruce, apod.), které obvykle nejsou primárně určena k uchycení na tělo [1, 2]. Nositelná zařízení současné generace jsou charakteristická tím, že se jedná o digitální systémy, které obsahují zabudované senzory a jsou schopna komunikovat bezdrátově s ji-

nými zařízeními [3].

Někdy jsou jako součást konceptu nositelných zařízení označována i implantovatelná zařízení [4]. Ta se od nositelných zařízení, jak jsou běžně chápána liší především ve způsobu připevnění na tělo, kdy jsou obvykle implantována přímo do těla (např. pod kůži ale i hlouběji). Stávají se tak nedílnou součástí člověka, který je nosí. Tyto značně miniaturizovaná zařízení jsou přitom stále schopna bezdrátové komunikace a to např. pomocí technologií RFID, respektive NFC [5].

Nositelná zařízení jsou v dnešní době žhavým tématem a zažívají velký rozmach. Jejich podíl na trhu se každým rokem zvyšuje, což odpovídá stále rostoucí

14

15

16

17

18

19

20

21

22

23

24

25

26

27 popťávce. Stejně tak jsou stále vymýšleny nové způ- 79
28 soby jejich aplikace. Mohou být pouhým módním do- 80
29 plňkem, sloužit k zábavě, nakupování, monitorování 81
30 osob, usnadnění práce a mnohým jiným praktickým 82
31 činnostem. V současné době nacházejí své největší 83
32 uplatnění v oblasti sportu, zdraví a životního stylu. Nej- 84
33 známějšími zástupci z této oblasti jsou fitness náramky 85
34 a chytré hodinky. 86

35 Tato práce se zaměřuje na popis využití nositel- 79
36 ných zařízení pro účely ovládání robota. V této oblasti 80
37 může být jejich nasazení vhodnější než použití jiných 81
38 způsobů ovládání. Problematicke se však nevěnuje 82
39 obecně, ale popisuje konkrétní řešení a to ovládání 83
40 pojízdného robota. Možností, jak přistoupit k tomuto 84
41 požadavku je vícero. Hlavním předpokladem je to, že 85
42 konkrétní nositelné zařízení, které je pro tuto úlohu 86
43 použité má obecnější charakter a disponuje senzory, 87
44 které dokáží rozpoznat pohyb zařízení, respektive těla 88
45 v prostoru. Tento pohyb je následně možné převést 89
46 na pohyb robota. Stejně tak použité nositelné za- 90
47 řízení musí disponovat rozhraním pro bezdrátovou ko- 91
48 munikaci, aby bylo schopné komunikovat s robotem, 92
49 nebo jiným zařízením. Jiným možným přístupem k 93
50 ovládání robota by bylo hlasové ovládání, kterým by 94
51 se však komplexnější pohyby robota daly řídit jen 95
52 stěží. Varianta to nicméně také je, ale v tomto pří- 96
53 padě se touto cestou nevydáme, protože aplikace na 97
54 rozpoznání hlasových povelů by byla mnohem složi- 98
55 tější, než aplikace na rozpoznání pohybů. Málokteré 99
56 nositelné zařízení je však natolik komplexní, aby bylo 100
57 možné přímo pro něj vyvinout aplikaci, která by řízení 101
58 robota zajistila. Dalším předpokladem tak bude to, že 102
59 pro ovládání robota bude využito mobilní zařízení na 103
60 platformě Android. 104

61 Finální řešení využívá nositelné zařízení s názvem 79
62 Myo armband (představeno v podkapitole 2.1 [6, 7, 8]). 80
63 Toto je ve tvaru náramku, jenž se uchytí na předloktí a 81
64 je schopný díky odpovídajícím senzorům zachytávat 82
65 pohyb v prostoru a také díky senzorům EMG (snímání 83
66 svalové aktivity) je schopný rozeznat základní gesta 84
67 ruky. S mobilním zařízením na platformě Android 85
68 komunikuje pomocí bezdrátové technologie Bluetooth 86
69 LE. Mobilní zařízení, respektive aplikace na něm běžící 87
70 (popsána v podkapitole 2.3) je poté na základě získa- 88
71 ných dat z náramku schopna komunikovat s robotem 89
72 (jsme s ním seznámeni v podkapitole 2.2 [9, 10, 11]). 90
73 Naše řešení využívá pro tuto komunikaci bezdrátovou 91
74 technologii Wi-fi a je takto schopné ovládat robota, 92
75 který běží na platformě ROS (Robotický operační sys- 93
76 téma). Samotné ovládání robota pomocí nositelného 94
77 zařízení Myo armband tak probíhá zprostředkovaně 95
78 pomocí mobilní aplikace. 96

V kategorii nositelných zařízení lehké váhy nemá 79
Myo armband příliš velkou konkurenci, zejména kvůli 80
jeho schopnosti rozeznávat už v základu určitá ge- 81
sta. Proto zde mnoho podobně použitelných řešení 82
nenalezneme. Co se týče kategorie nositelných za- 83
řízení těžké váhy, tak zde existují např. nositelné exo- 84
skeletony, které detekují pohyb na základě dat z po- 85
tenciometrů a umožňují kontrolovat bezdrátově (např. 86
pomocí Bluetooth) pohyby humanoidního robota [12]. 87
V oblasti ovládání humanoidních robotů tak nemůže 88
mít tento přístup vážnější konkurenci. 89

90 Naše řešení založené na Myo armband a mobilní 91
92 aplikaci se však nezaměřuje na humanoidní roboty, 93
94 ale pojízdné roboty. Toto řešení je poměrně obecné, 95
96 protože není vytvořeno pro konkrétní model robota, 97
98 ale je možné jej aplikovat na celou řadu robotů, které 99
100 běží pod ROS. Také je mobilní a přenositelné, pro- 101
102 tože náramek máme upevněný na ruce a mobil mů- 103
104 žeme držet v jedné ruce nebo si jej strčit do kapsy, 105
106 klidně si tak můžeme vyjít s robotem na procházku. 107
108 Hlavní nevýhoda je ta, že dosah technologie Wi-fi není 109
110 tak velký jako dosah klasického rádiového vysílače. 111
112 Náramek Myo je poměrně čerstvou záležitostí, ale už 113
114 nyní vznikají jeho aplikace pro ovládání robotů. Na 115
116 Google Play je zatím skutečně několik málo aplikací, 117
118 které slouží pro ovládání robotů (většinou dronů), za- 119
120 tím ale trpí nedostatkem obecnosti a na spojení ROS 121
122 a Myo náramku jsem na Google Play nenarazil. Ne- 123
124 oficiálně ale samozřejmě mohou existovat podobné 125
126 aplikace. 127

2. Architektura systému a jeho kompo- 109 nenty 110

111 Celý systém, jak již bylo naznačeno se skládá ze tří 112
113 komponent a to nositelného zařízení Myo armband, 114
115 mobilní aplikace na platformě Android a robota na 116
117 platformě ROS. Jejich vzájemnou komunikaci ilustruje 118
119 lépe obrázek 1. Samotné řešení, tedy vlastní přínos, 120
121 pak spočívá v mobilní aplikaci, která přijímá data z 122
123 Myo náramku a na základě nich zasílá řídící zprávy 124
125 robotovi. Komunikace náramek → mobil využívá tech- 126
127 nologii Bluetooth LE, zatímco komunikace mobil → 128
129 robot používá technologii Wi-fi. Jednotlivé kompo- 130
131 nenty systému a s nimi související technické detaily 132
133 jsou představeny v následujících podkapitolách. 134



Obrázek 1. Architektura systému (popisuje komponenty a způsob komunikace).

Indikační led diody Dvě světelné diody, které indikují stav zařízení (např. nabíjí se, synchronizováno, zahřívá se, aj.).

Haptickou zpětnou vazbu Jinými slovy se jedná o vibrátor, který slouží jako zpětná vazba (např. při detekování nového gesta ruky). Je možné jej ovládat softwarově.

Rozhraní Bluetooth LE Slouží ke komunikaci s jinými zařízeními. Pomocí něj zpřístupňuje data ze senzorů, gesta, stav baterie, aj.

Rozhraní Micro USB Slouží k zapnutí zařízení a k nabíjení zabudované Lithium-Ion baterie, která na jedno nabití vydrží v nepřetržitém provozu celý den.

2.1 Nositelné zařízení Myo armband

Myo armband je nositelné zařízení ve tvaru náramku, které se nosí na předloktí a disponuje nejrůznějšími senzory, které umožňují zachytit pohyby ruky a některá gesta, což je velmi zajímavá schopnost. Mezi jeho typické aplikace patří ovládání techniky (HW) nebo aplikací (SW), typicky se tak uplatňuje při ovládání počítačových her, aplikačních programů (prezentace, video, hudba, aj.), robotů, ohňostrojų nebo ve virtuální realitě. Výrobcem je kanadská společnost Thalmic Labs, která byla založena v roce 2012. Společnost sebe samu definuje jako průkopníka v oblasti nových způsobů interakce s digitálními zařízeními.

Co se týče provedení, náramek existuje v bílém nebo černém provedení (viz obrázek 2). Je schopen přizpůsobit se různým obvodům předloktí (19 cm až 34 cm). Váha je snesitelných 93 g a tloušťka 1.14 cm. Jeho hardwarová výbava zahrnuje:

EMG senzory Elektromyografické senzory, jejichž část, která je v kontaktu s kůží, je vyrobena z chirurgické oceli. Senzory snímají elektrickou svalovou aktivitu v předloktí a na základě dat z těchto senzorů je náramek schopen rozpoznat některá základní gesta ruky (viz obrázek 3). Celkově obsahuje náramek po svém vnitřním obvodu 8 EMG senzorů. Data ze senzorů EMG jsou snímána s frekvencí 200 Hz.

IMU jednotku Vysoce citlivá inerciální měřicí jednotka, která zahrnuje tří-osý gyroskop, tří-osý akcelerometr a tří-osý magnetometr. Na základě dat z těchto senzorů je zařízení schopné detekovat pohyb v prostoru včetně rotace (viz obrázek 4). Data ze senzorů IMU jsou snímána s frekvencí 50 Hz.

Procesor ARM Cortex M4



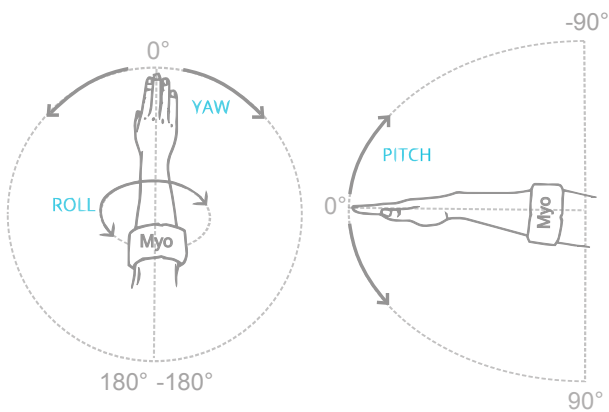
Obrázek 2. Nositelné zařízení Myo armband v tmavém provedení. (Zdroj: [13])



Obrázek 3. Základní gesta detekovatelná zařízením Myo armband. Zleva: *Fingers Spread*, *Fist*, *Double Tap*, *Wave Left* a *Wave Right*. (Zdroj: [13])

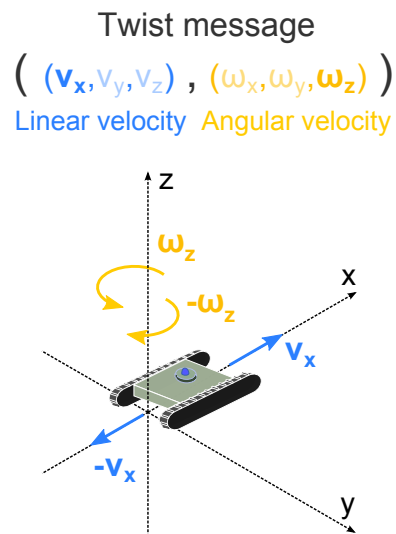
Mezi podporované platformy pro které existuje vývojářské API patří Android, Mac OS X, iOS (vybrané modely zařízení iPod, iPad a iPhone) a Windows. Přičemž Mac OS X a Windows potřebují ke komunikaci s Myo armband výrobcem dodávaný USB adaptér Bluetooth LE. Pro ostatní platformy (např. Windows Phone) neexistuje vývojářské API a je nutné řešit komunikaci s Myo armband na nižší úrovni v rámci rozhraní Bluetooth LE.

Jelikož je možné číst z náramku surová EMG data zachycující svalovou aktivitu (nejen předem definovaná gesta, která jsou rozpoznána na základě svalové aktivity), tak je možné softwarově implementovat vlastní rozpoznání gest. Úkol je to nicméně nesnadný a pro mnoho aplikací postačí předem definovaná gesta.



Obrázek 4. Pohyby v prostoru, které je schopné zařízení díky svým senzorům detekovat. Pohyby typu YAW a PITCH naše aplikace využívá pro ovládání robota. Více informací v kapitole 3. (Obrázek vytvořen na základě zdroje: [14])

Jedním z požadavků na robota je i to, aby podporoval řídicí zprávy typu Twist. Účel a strukturu těchto zpráv popisuje obrázek 5.



Obrázek 5. Řídicí zpráva sestává ze dvou třísložkových vektorů. První vektor představuje lineární rychlost. V tomto vektoru nás zajímá pouze jeho složka v_x , která určuje lineární rychlost robota (jednotky m/s) v třírozměrném prostoru ve směru osy x . Robot se tak pohybuje dopředu (kladná hodnota v_x), nebo dozadu (záporná hodnota v_x). Druhý vektor představuje angulární rychlost. V tomto vektoru nás zajímá pouze jeho složka ω_z , která určuje angulární rychlost robota (jednotky rad/s) v třírozměrném prostoru kolem osy z . Robot se tak otáčí doleva (kladná hodnota ω_z) nebo doprava (záporná hodnota ω_z).

Pro demonstraci použitelnosti aplikace je tato aplikace předváděna s reálným robotem, na kterém proběhlo také její hlavní otestování (průběžné testy používaly simulátor robota). Tímto robotem je fakultní pojízdný (a díky dobíjecí baterii také vysoce mobilní) robot s názvem Trilobot (viz obrázek 6). Disponuje třemi koly, přičemž pohon má na dvě z nich. Tento pohon je zajištěn dvěma sériovými motory s převodovkami. Robot je řízen mikropočítači Odroid a Arduino MEGA, kde:

Odroid je použit jako hlavní řídicí jednotka robota, která jej ovládá na vyšší úrovni. Je na něm nainstalováno Ubuntu a v rámci Ubuntu metaoperační systém ROS verze Indigo. Je vybaven také USB Wi-fi kartou, pomocí které může komunikovat bezdrátově s jinými zařízeními (v našem případě s mobilní aplikací). Také je nakonfigurován tak, aby se automaticky po zapnutí připojil k předem nastavenému Wi-fi AP. Není

2.2 Robot

Pomocí nositelného zařízení Myo armband a mobilní aplikace je možné ovládat velkou množinu robotů. Kompatibilní robot musí mít následující parametry:

- Je pojízdný ve smyslu pohybu vpřed, vzad a zatáčení.
- Je poháněn Robotickým operačním systémem.
- Podporuje přijímání řídicích zpráv typu Twist (v rámci ROS), které řídí pohyb robota a disponuje rozhraním Wi-fi, pomocí kterého jsou mu tyto řídicí zprávy zasílány.

Robota s výše uvedenými parametry jsme pak schopni pomocí náramky Myo a mobilní aplikace ovládat. Jedním ze základních požadavků je, aby byl poháněn Robotickým operačním systémem (ROS). Tento projekt vznikl v roce 2007 ve Stanford Artificial Intelligence Laboratory. Jedná se o open-source meta-operační systém/framework určený pro roboty. Poskytuje hardwarovou abstrakci, implementaci často požadované funkcionality, předávání zpráv mezi procesy, nástroje a knihovny pro psaní a spouštění kódu mezi více zařízeními. Cílem je zjednodušit vytváření složitých robotických aplikací napříč širokým spektrem robotických platforem. Grafem běhového prostředí ROS je síť peer-to-peer procesů/uzlů, které mohou běžet distribuovaně napříč více stroji. ROS umožňuje několik stylů komunikace mezi procesy/uzly, např. synchronní vzdálené volání procedur prostřednictvím služeb nebo asynchronní přenos dat prostřednictvím topiců. Naše řešení funguje právě na bázi asynchronní komunikace, kdy mobilní aplikace představuje proces/uzel, který publikuje zprávy do topicu, který je součástí procesu/uzlu běžícím na robotu.

potřeba jej tak po každém zapnutí znovu konfigurovat. Na Odroidu běží ROS program, který představuje uzel a naslouchá na topicu s názvem `/cmd_vel_safe`. Ovládání robota zvenčí se děje zasíláním zpráv typu `geometry_msgs/Twist` přes Wi-fi na tento topic, který slouží k ovládání pohonu kol.

Arduino MEGA je použito pro ovládání robota na nižší úrovni. K Odroidu je připojen pomocí USB rozhraní. Běží na něm podobně jako na Odroidu ROS program, který představuje uzel a naslouchá na topicu s názvem `/trilobot/cmd_vel` a očekává zprávy typu `geometry_msgs/Twist`. Tyto zprávy získává přes USB rozhraní od Odroidu. Na základě těchto zpráv ovládá motory, které jsou k němu připojeny pomocí V/V pinů.

Pokud robot nezachytí zprávu déle jak 50 ms, tak se sám zastaví. Funkce robota a jeho řídicích komponent jsou ve skutečnosti ještě o něco složitější a robot toho umí více. My jsme se zde ale zaměřili jen na to, co bezprostředně souvisí s naším problémem řízení pohybu robota.



Obrázek 6. Fakultní pojízdný robot s názvem Trilobot.

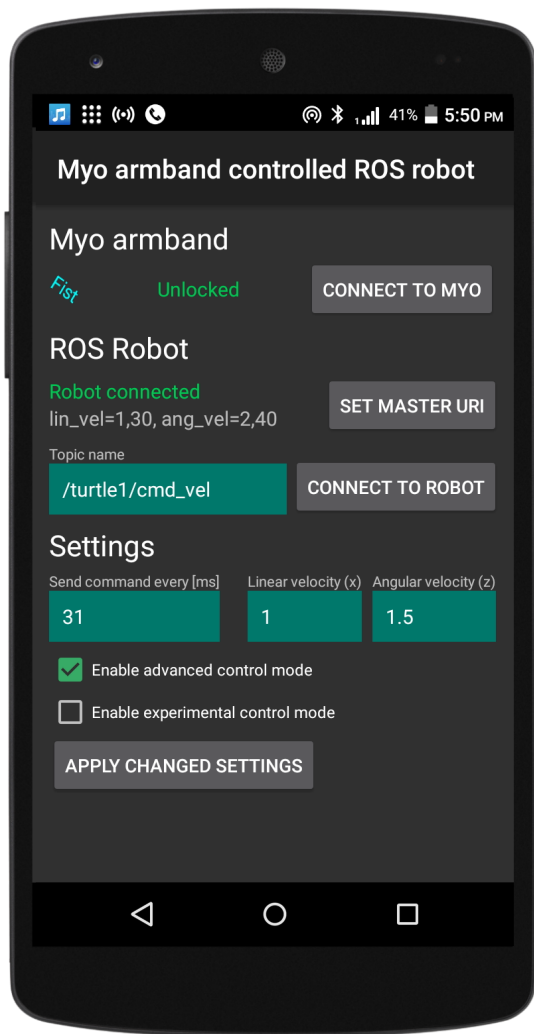
2.3 Mobilní zařízení a aplikace

Mobilní aplikace je jádrem celého řešení a výsledného systému. Aplikace vyžaduje pro svůj chod mobilní zařízení s verzí Android 4.3 a vyšší a také s podporou technologie Bluetooth LE. V této podkapitole se zaměříme nikoliv na implementační část aplikace (ty řeší kapitola 3), ale na možnosti aplikace a způsob jejího použití a propojení všech tří klíčových komponent (Myo náramek, robot, aplikace) do výsledného systému. Hlavní obrazovku hned po startu aplikace zobrazuje obrázek 7. Můžeme si na něm povšimnout že se dělí na 3 části:

1. **Myo armband** – odpovídá Myo náramku. Můžeme zde sledovat stav náramku, tedy jestli je

spárován s aplikací, nebo je odpojený. Můžeme se k náramku připojit pomocí tlačítka **CONNECT TO MYO**, popř. se takto od něj odpojit. Pokud je náramek odpojený, je tato informace signalizována pomocí červeného textu **Disconnected**. Můžeme také sledovat, jestli je náramek odemčený **Unlocked** nebo zamčený **Locked**. Gestu ruky se rozpoznávají pouze, když je náramek odemčený a tato rozpoznaná gesta aplikace zobrazuje (v tomto konkrétním případě je aplikace zachycena při rozpoznání **Fist**). Pomocí pohybu náramku se text popisující rozpoznaná gesta různě natáčí, čímž také můžeme kontrolovat, jestli se data o poloze náramku v prostoru správně přenáší.

2. **ROS Robot** – odpovídá robotu. Můžeme zde sledovat, jestli je aplikace připojena k robotu, respektive jestli v pořádku funguje zasílání řídicích zpráv ROS uzlu/procesu pomocí daného topicu. Pakliže je zasílání zpráv funkční, je tato informace signalizována pomocí textu **Robot connected** a také jsou k dispozici data (hodnota lineární rychlosti `lin_vel` a angulární rychlosti `ang_vel`), která jsou obsahem zpráv. V opačném případě je zobrazen červený text **Robot disconnected**. Připojit se k robotu, respektive spustit proces zasílání řídicích zpráv ROS uzlu/procesu na robotu můžeme pomocí tlačítka **CONNECT TO ROBOT**. Předtím je ale nutné nastavit **MASTER URI** (většinou IP adresa robota, respektive IP adresa WLAN interface robota přes které je spojen s aplikací + číslo portu na kterém běží **ROS core**). Také zde nastavujeme název topicu na robotovi, do kterého bude aplikace publikovat řídicí zprávy (v tomto případě `/turtle1/cmd_vel`).
3. **Settings** – odpovídá nastavení aplikace. Můžeme zde nastavit časový interval zasílání řídicích zpráv (v tomto případě 31 ms), protože různí roboti vyžadují různé maximální časové intervaly. Také zde můžeme nastavit základní lineární rychlost (v tomto případě 1 m/s) a základní angulární rychlost (v tomto případě 1.5 rad/s). Další, co zde můžeme nastavit, jsou různé módy ovládání. Tyto módy ovlivňují, jakými gesty a pohyby ruky můžeme robota ovládat a jak na takové ovládání reaguje. Více o módech a jím odpovídajících pohybech je uvedeno v kapitole 3. Aplikace si také pamatuje naposledy nastavené hodnoty co se týče nastavení, ale i název topicu a **MASTER URI**; při příštím spuštění tak nemusíme vše nastavovat opět znovu.



Obrázek 7. Aplikace na platformě Android, která řídí robota na základě dat z náramku Myo armband.

vlákna, ale ne vláken běžících na pozadí. Ty se ukončí, 347
pokud se skutečně ukončí celá aplikace. 348

Jednotlivé třídy: 349

SettingsSingleton Implementuje návrhový vzor Sin- 350
gleton. Obsahuje členské proměnné: *linearVe-* 351
locityX (lineární rychlost robota), *angularVeloc-* 352
ityZ (angulární rychlost robota) a *sleepTime* (in- 353
terval posílání řídicích zpráv robotovi). Všechny 354
tyto proměnné jsou tzv. volatilní, takže jejich 355
změna je viditelná okamžitě napříč vlákny a 356
slouží tak k mezivláknové komunikaci. Rych- 357
lostní proměnné nastavuje vlákno, které získává 358
data z Myo náramku a jiné vlákno, které se 359
stará o zasílání řídicích zpráv robotovi hodnoty 360
těchto proměnných čte a používá v zasílaných 361
zprávách. 362

ROSMessagesenderNode Představuje implementaci 363
ROS uzlu/procesu, který je vykonáván v tzv. 364
vlákně běžícím na pozadí. Jeho hlavní činnost 365
spočívá v periodickém zasílání řídicích zpráv 366
typu *Twist* prostřednictvím topicu uzlu/procesu 367
běžícím na robotu. Řídicí zpráva (blíže popsána 368
na obrázku 5), sestává ze dvou vektorů typu 369
Vector3. Složku v_x vektoru lineární rychlosti 370
naplníme hodnotou proměnné *linearVelocityX* 371
z objektu třídy *SettingsSingleton*. Ostatní jeho 372
složky nastavíme na hodnotu 0. Složku ω_z vek- 373
toru angulární rychlosti naplníme hodnotou pro- 374
měnné *angularVelocityZ* z objektu třídy *SettingsS-* 375
ingleton. Ostatní jeho složky nastavíme na hod- 376
notu 0. 377

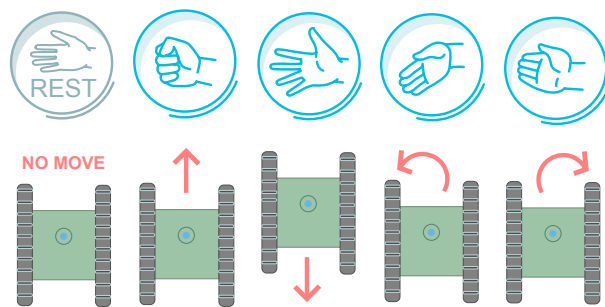
MainActivity Představuje v řeči Android tzv. hlavní 378
aktivitu aplikace. Zde se nacházejí metody pro 379
vykreslování GUI, obsluhu jeho událostí a spou- 380
štění jiných aktivit. Obsahuje vnořenou třídu 381
DeviceListener, jejíž metody představují asyn- 382
chronní rozhraní, pomocí kterého jsme schopni 383
získávat data z náramku Myo. Nejklíčovější 384
jsou metody *onOrientationData()* a *onPose()*. 385
Metoda *onPose()* se zavolá vždy, když náramek 386
zaregistruje nové gesto ruky. Na základě poskyt- 387
nutého gesta, aktuálního ovládacího módu a pří- 388
padně stavových proměnných provádíme akce 389
jako např. nastavení angulární a lineární rychlosti. 390
Metoda *onOrientationData()* se zavolá vždy při 391
poskytnutí nových orientačních dat (zhruba ka- 392
ždých 50 ms). Z těchto orientačních data získá- 393
váme Eulerovy úhly – *yaw* (úhel precese), *pitch* 394
(úhel nutace) a *roll* (úhel rotace), které před- 395
stavují dílčí rotace vzniklé rozkladem obecné 396
prostorové rotace [15]. Díky nim jsme schopni 397
určit pohyb a polohu náramku v prostoru. Os- 398

328 3. Implementace a způsob ovládání

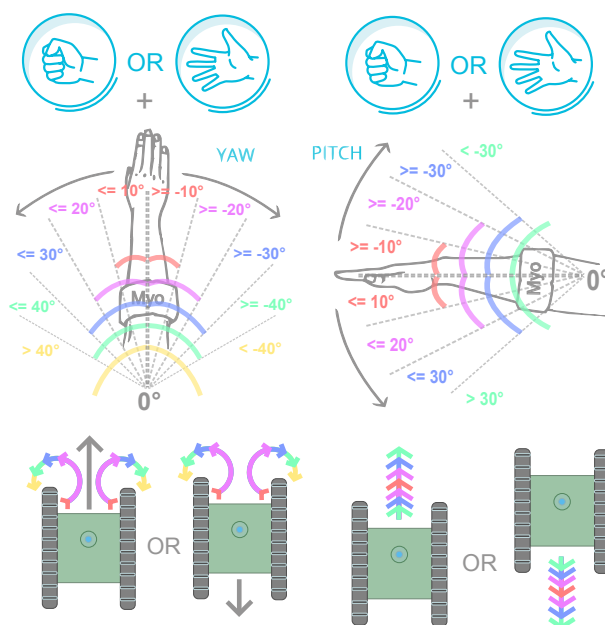
329 Implementace byla realizována v jazyce Java na plat-
330 formě Android. Přitom bylo využito knihoven třetích
331 stran, které implementaci odstínilo od skutečně níz-
332 kourovňových záležitostí, zejména bezdrátové komu-
333 nikace. Konkrétně bylo využito tzv. *Myo SDK*, což
334 je sada knihoven, které poskytují API pro komunikaci
335 s Myo náramkem. Také bylo využito experimentál-
336 ních knihoven *ROSjava*, které poskytují API pro ko-
337 munikaci s ROS robotem. Obě API jsou asynchronní.

338 Vlastní řešení se skládá ze tří souborů: *MainActiv-*
339 *ity.java*, *ROSMessagesenderNode.java* a *SettingsSin-*
340 *gleton.java*, které představují samostatné třídy; jejich
341 účel bude představen záhy. Aplikace kromě hlavního
342 vlákna také sestává ze dvou vláken běžících na pozadí –
343 jedno vlákno pro obsluhu náramku, druhé pro obsluhu
344 robota. Tato vlákna spolu komunikují. Díky tomuto
345 mechanismu funguje ovládání robota i pokud spustíme
346 jinou aplikaci, protože se přeruší vykonávání hlavního

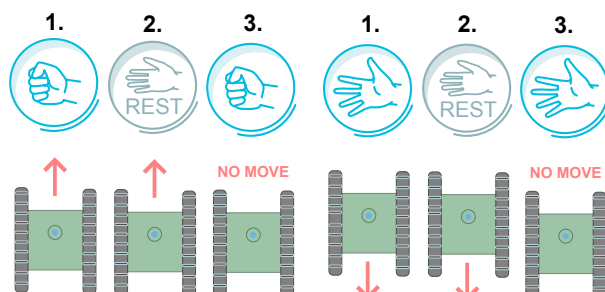
tatně toto bylo již předmětem obrázku 4. Aplikace využívá úhly *yaw* a *pitch*. Rozsah *yaw* jak je patrné také z obrázku 4 je 360° (0° až -180° a 0° až 180°). Využíváme jej pro ovládání zatáčení robota. Rozsah *pitch* jak je patrné také z obrázku 4 je 180° (-90° až 90°). Využíváme jej pro regulaci/zvýšení rychlosti robota. Využití orientačních dat je závislé na módu ovládání robota, více o něm dále.



Obrázek 8. Základní mód, kdy je robot ovládán pomocí čtyř gest. (Obrázek vytvořen na základě zdroje: [13])



Obrázek 9. Pokročilý mód, který umožňuje ovládání pohybem ruky během jízdy vpřed nebo vzad. Robot tak může za jízdy zatáčet a zvyšovat/snižovat rychlost. (Obrázek vytvořen na základě zdrojů: [14, 13])



Obrázek 10. Experimentální mód šetří naše svaly, aby robot jel dopředu nebo dozadu, stačí provést gesto jedenkrát a není nutné jej držet po celou dobu pohybu. (Obrázek vytvořen na základě zdroje: [13])

Práce v úvodu pojednávala o nositelných zařízeních a jejich možném využití pro ovládání robota, včetně představení konkrétního již realizovaného řešení. Toto řešení pak bylo v dalších kapitolách více rozvíjeno. Byla popsána jeho architektura a její komponenty. Zde jsme se dozvěděli, že celý systém ovládání robota pomocí nositelného zařízení sestává ze tří klíčových komponent. První komponentou je nositelné zařízení Myo armband, které je vybaveno senzory jako akcelerometr, gyroskop, magnetometr, které jsou schopné detekovat polohu a pohyb zařízení v prostoru a EMG senzory, které jsou schopné detekovat svalovou aktivitu a na základě nich zařízení umožňuje detekovat základní gesta ruky. Další komponentou je mobilní aplikace na platformě Android, která pomocí technologie Bluetooth LE čte data ze senzorů a na základě nich pomocí algoritmu zasílá prostřednictvím technologie Wi-Fi řídicí zprávy třetí komponentě a to robotovi, který je poháněn Robotickým operačním systémem. V další kapitole byla detailněji rozebrána implementace aplikace a také způsob ovládání robota.

Hlavním výsledkem práce je Android aplikace určena pro ovládání pojízdných robotů na platformě ROS. Kromě aplikace je také třeba disponovat nositelným zařízením Myo armband. Kombinace těchto dvou komponent pak vytváří zajímavý, obecný, flexibilní, přenositelný a především prakticky použitelný systém, který je schopen řídit širokou množinu robotů. Demonstrace použití tohoto systému při ovládání reálného robota je zachycena na videu v rámci přiložených materiálů na začátku článku.

Přínos článku spočívá v seznámení s nositelnými zařízeními a jejich použitím pro ovládání robota včetně vytvořené aplikace, kterou lze pro tyto účely prakticky použít. Čtenář si z článku může odnést dojem o tom, jaké existují současné způsoby pro ovládání robotů.

Vývojáři a výzkumníci mohou použít představenou aplikaci pro ovládání např. svého pojízdného robota, který běží na platformě ROS (musí si však také pořídit daný náramek). Také mohou aplikaci rozšířit za účelem ovládání i dalších typů ROS robotů. Zatím jediný můj záměr je publikovat tuto aplikaci na Google Play a zpřístupnit tak širší veřejnosti.

489 Poděkování

490 Chtěl bych poděkovat mému vedoucímu práce Dr.
491 Janu Samkovi a také Ing. Radimu Lužovi a Dr. Jaroslavu
492 Rozmanovi za jejich odbornou pomoc.

- [1] Wearable technology [online]. <http://goo.gl/Is27BF>, ©2015 [cit. 2015-12-03]. 494 495
- [2] Wearables: nositelná elektronika [online]. <http://goo.gl/5Zu5qn>, ©2012 - 2014 [cit. 2015-12-03]. 496 497 498
- [3] What is wearable tech? everything you need to know explained [online]. <http://goo.gl/ueokRU>, 2014 [cit. 2015-12-03]. 499 500 501
- [4] Tehrani Kiana and Andrew Michael. Wearable technology and wearable devices: Everything you need to know [online]. <http://goo.gl/tmvvtw>, Updated: March 26, 2014 [cit. 2015-12-03]. 502 503 504 505 506
- [5] Murray Slovic. Implantable microchips are the ultimate wearable [online]. <http://goo.gl/tmvvtw>, JANUARY 07, 2015 [cit. 2015-12-03]. 507 508 509
- [6] Tech specs [online]. <https://goo.gl/IYFmI4>, ©2013 - 2016 [cit. 2016-03-24]. 510 511
- [7] Myo armband [online]. <https://goo.gl/dsq8mR>, Last modified on 1 January 2016 [cit. 2016-03-24]. 512 513 514
- [8] Emg sampling rate [online]. <https://goo.gl/vk5cla>, March 7, 2015 [cit. 2016-03-25]. 515 516
- [9] Radek Vopálenický Pavel Hájek, Zdeněk Hartman. Modernizace trilobota [el. dokument], 23. prosince, 2015 [cit. 2016-03-25]. 517 518 519
- [10] Thomas Dirk. Ros introduction [online]. <http://goo.gl/aXomsE>, Naposledy editováno 2014-05-22 [cit. 2016-03-25]. 520 521 522
- [11] Linear and angular velocity [online]. <http://http://goo.gl/xiwHgc>, ©2003-2016 [cit. 2016-03-25]. 523 524 525
- [12] Liat Clark. Watch this lego designer control a robot with his exoskeleton [online]. <http://goo.gl/bRzkgu>, 29 July 15 [cit. 2016-04-03]. 526 527 528 529
- [13] Tech specs [online obrázek]. <https://goo.gl/IYFmI4>, ©2013 - 2016 [cit. 2016-03-25]. 530 531
- [14] Mark Difrancio. Gui without the g: Going beyond the screen with the myo armband [online obrázek]. <http://goo.gl/lS4SFH>, Mar 11th, 2014 [cit. 2016-03-30]. 532 533 534 535
- [15] Eulerův úhel [online]. <http://goo.gl/jRPsgJ>, ©2012 [cit. 2016-04-02]. 536 537