

TAČR TREND

Název projektu:

Flexibilní robotické pracoviště pro malosériovou výrobu
(FW03010501)

Technická zpráva

[O] Koncepce programové podpory zpracování naučených pohybů robotu

Martin Švejda (ZČU), Arnold Jáger (ZČU)

datum: 29. 12. 2021



Identifikátor účastníka projektu:

IMI: Integrated Micro-Electronics Czech Republic s.r.o.

ZČU: Západočeská univerzita v Plzni

Obsah

Popis z přihlášky projektu	2
Úvod	3
Datový postprocesor (proof of concept)	3
Replikace naučeného pohybu	5
Aproximace naučeného pohybu s řízením profilu rychlosti	7
Video demonstrace	8
Reference	8

Popis z přihlášky projektu

ZČU: Koncepce SW aplikace pro zpracování zájmových bodů, které operátor definuje prostřednictvím navádění robotu v rámci požadované činnosti. ZČU předkládá možné metody zpracování zájmových bodů (standardní používané pohyby - lineární, kruhový, atd.) + nové možnosti interpolace komplexních pohybů - které nelze často běžnými komerčními systémy interpretovat.

IMI: Definuje standardní požadované úkony dosud manuálně ovládaných pracovišť s ohledem na geometrie pohybů (zejména při nestandardních aplikacích jako např. zapojování konektorů, ukládání diskrétních součástí, atd.). Tyto požadavky se mohou zásadně lišit od požadavků na běžné pick&place aplikace.

Úvod

Technická zpráva [2] představuje koncepci řešení pro intuitivní navádění robotu (Staubli, UR, Fanuc) prostřednictvím přidaného senzoru v podobě 3D myši. Díky takovému řešení může operátor do jisté míry převzít kontrolu nad robotem a na místo pohybu s robotem pomocí teachpendantu přímo robot ovládat tažením za 3D myš a to jak v případě uchycení myši ke koncovému efektoru robotu tak v podobě dálkového ovládání, kdy je myš umístěna na konzoli mimo robot.

Otázkou zůstává, jak takový systém využít k plánování pohybu robotu ve smyslu naučení zájmové trajektorie operátorem. Principiálně lze uvažovat dvě možnosti:

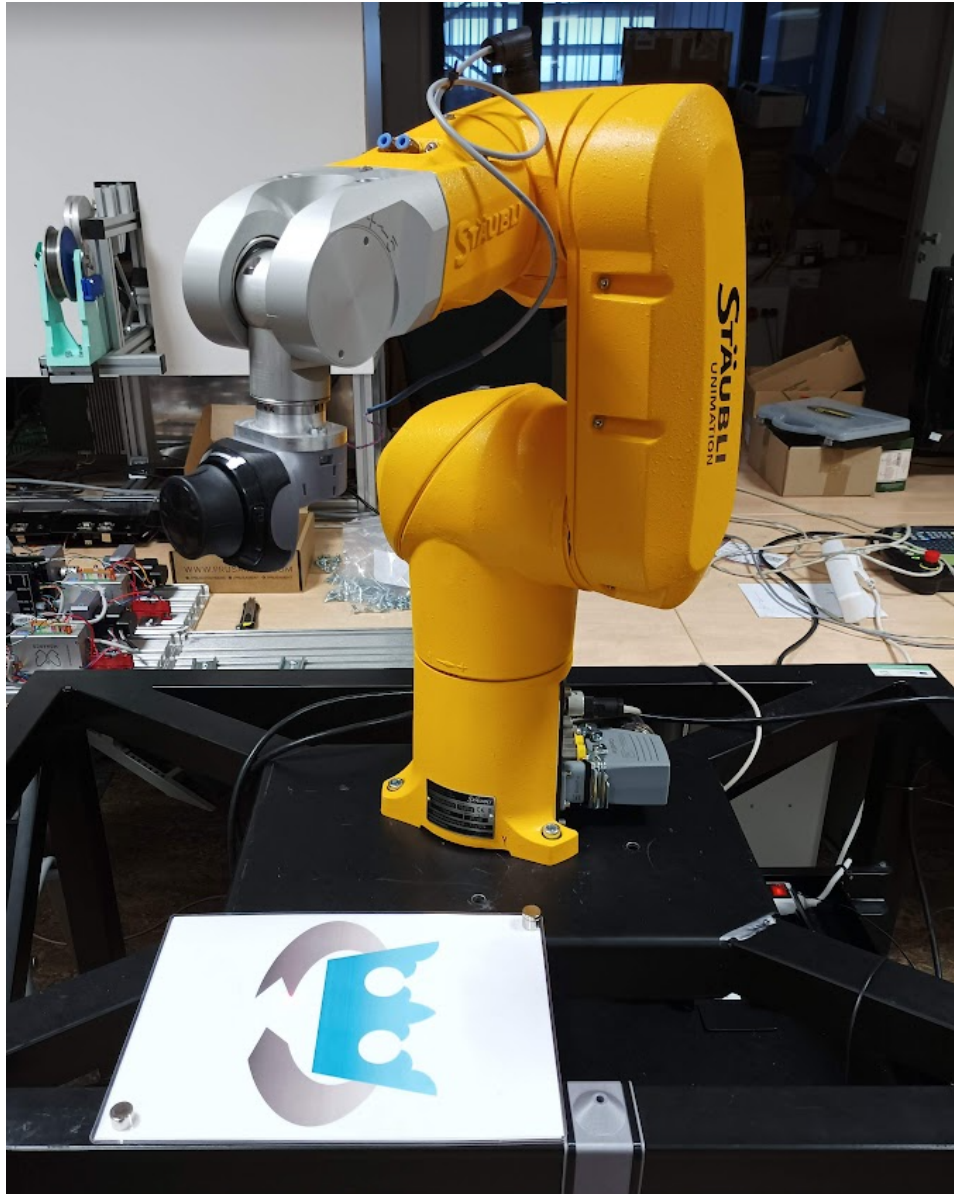
1. Během učení pohybu operátorem tažením za 3D myš ukládat diskrétní (zájmové) polohy robotu a tyto polohy poté využít v originálním kontroleru robotu jako danou množinu naučených bodů. Takový přístup je přímočarý a nahrazuje nutnost takovou množinu bodů definovat prostřednictvím teachpendantu, kde samotné ovládání robotu nemusí být dostatečně intuitivní a vyžaduje jistou znalost práce s roboty.
2. Uložené datové body během procesu učení v podobě diskrétních bodů či kontinuálního záznamu trajektorie dále zpracovat v samostatné aplikaci - datovém postprocesoru a výsledný pohyb zpětně replikovat robotem, takový přístup představuje následující možnosti:
 - Využít nové a pokročilé metody interpolace diskrétních zaznamenaných bodů pro generování specifické pohybové trajektorie (např. s důrazem na hladkost pohybu)
 - Generovat komplexní pohybové trajektorie využívající zkušenosti operátora v dané aplikaci při kontinuálním záznamem pohybu (např. nanášení lepidla, laserové kalení, značení, atd.).
 - Provádět další úpravy nasnímané pohybové trajektorie - např. ruční korekce tvaru, profilu rychlosti posunu podél trajektorie, atd.

Druhá z uvedených možností byla podrobněji zkoumána a byla vytvořena samostatná aplikace v prostředí Matlab, která je podrobně popsána v následující kapitole.

Datový postprocesor (proof of concept)

Jako “proof of concept” datového postprocesoru naučených dat byla zvolena implementace aplikace v programu Matlab, která zpracovávala kontinuálně zaznamenané body během tažení robotu prostřednictvím 3D myši za koncový efektor.

Použitým robotem byl průmyslový robot Staubli TX40 s HW rozšířením uniVAL drive, viz Obrázek 1, jehož prostřednictvím komunikoval kontroler robotu s nadřazeným řídicím systémem (průmyslový počítač B&R). Nadřazený řídicí systém realizoval kompletní řízení pohybu robotu včetně integrace kontroleru 3D myši, více viz [2].



Obrázek 1: Robot Staubli s nadřazeným řídicím systémem implementovaným v SW REXYGEN s integrovaným kontrolerem 3D myši

Samotný datový postprocesor byl vytvořen v programu Matlab, který běžel na hostovském počítači (standardní PC, notebook) s operačním systémem Windows. Datový postprocesor se skládal z následujících částí:

- **Konfiguračního souboru (*configTrajGenerator.m*):** Obsahuje nastavení postprocesoru dat, konkrétně se jedná o parametry typu:
 - identifikace cílového počítače s řídicím systémem robotu (IP adresa, přihlašovací údaje, umístění datových souborů, atd.)
 - kinematické parametry robotu a omezení na pohyb kloubů (polohy, rychlosti, zrychlení)
 - parametry nastavení filtrů pro zpracování kontinuálně naučených dat tažením za 3D myš (nekausální FIR filtr, B-Spline filtr s řízením profilu rychlosti podél trajektorie)
 - a některé další...

- **Spustitelného skriptu (*setup_REX.m*):** Spuštěním skriptu v Matlabu se aktivuje nekonečná smyčka datového postprocesoru, která principem hand-shake komunikuje s řídicím počítačem robotu a dle požadavků aktivuje dílčí programové části postprocesoru dat (např. stažení/nahrání datových souborů na cílový řídicí počítač, zpracování dat filtrací dle zadaných parametrů, vyhodnocení realizovatelnosti výsledné pohybové trajektorie, atd.).
- **Dalších funkcí, které implementovaly dílčí programové části:** Typicky algoritmy filtrací, výpočtu inverzní kinematiky robotu, čtení a zápis dat mezi datovým postprocesorem a řídicím systémem robotu (použito rozhraní RESTful API).

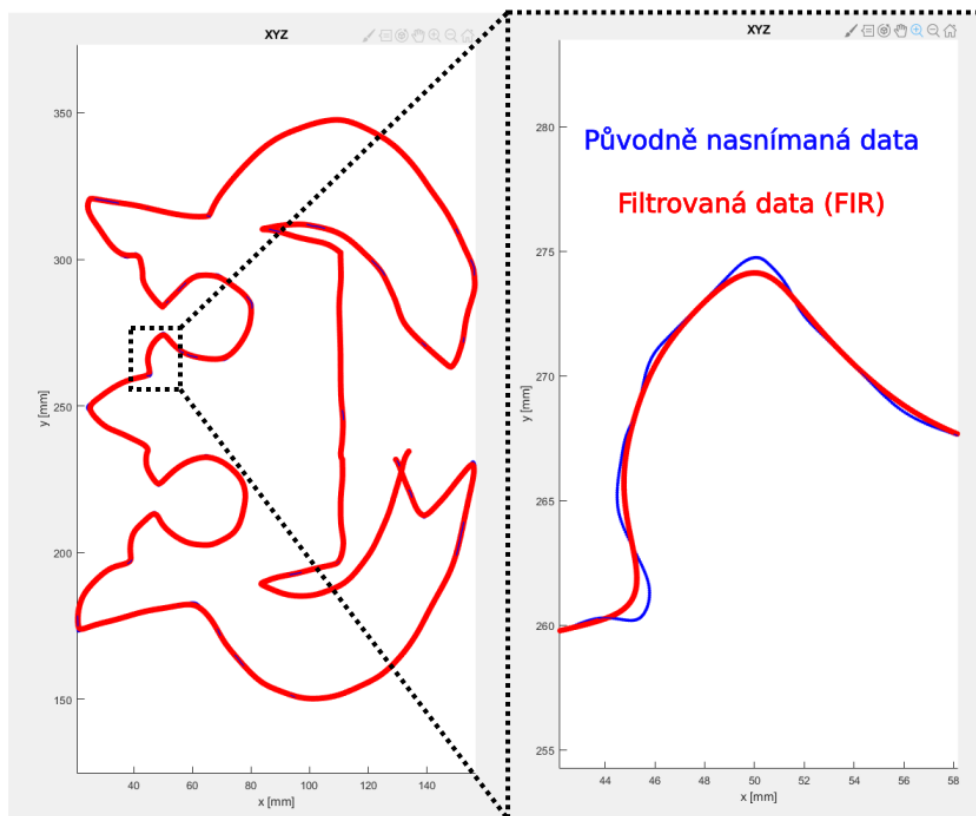
Datový postprocesor byl implementován pro dva režimy zpracování naučených dat, jak je uvedeno v následujících kapitolách.

Replikace naučeného pohybu

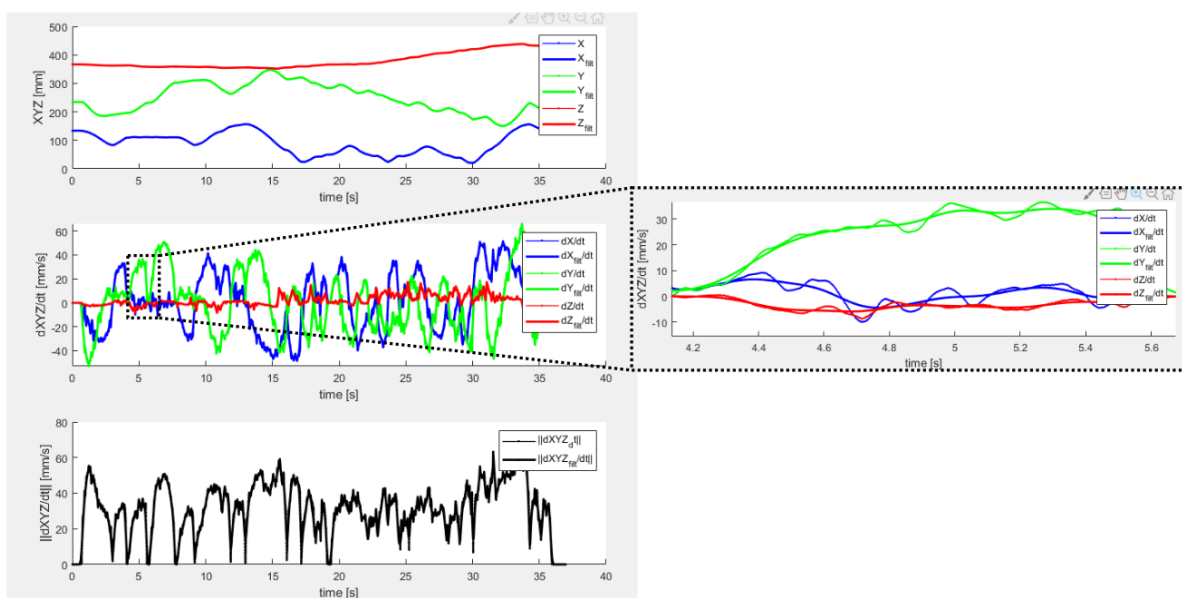
Replikace naučeného pohybu byla založena na využití nekauzální filtrace filtry s konečnou impulzní odezvou. Obrázek 2 a 3 demonstruje filtraci polohového signálu zaznamenaného při učení pohybu robotu operátorem v prostoru XYZ a v časové oblasti (uvažována jen translace). Filtrace signálu použitým filtrem vykazuje následující vlastnosti:

- FIR¹ filtr byl implementován jako digitální okénkový (Blackman–Harris) filtr, který neobsahuje kmitavé módy => filtrovaná trajektorie se nerozkmitává při rychlém vybuzení (rychlé přechody)
- Filtr nezavádí fázové zpoždění
- Šířkou okénka filtru lze měnit míru filtrace zaznamenaných dat - vyhlazení dat na úkor přesnosti a naopak
- Algoritmus je výpočetně efektivní a rychlý
- Filtrovaná data přesně kopírují profil rychlosti podél trajektorie (tzn. reflektují rychlost pohybu operátora při učení pohybu)

¹ Final Impulse Response - filtr s konečnou impulzní odezvou.



Obrázek 2: XYZ poloha zaznamenaných a filtrovaných dat

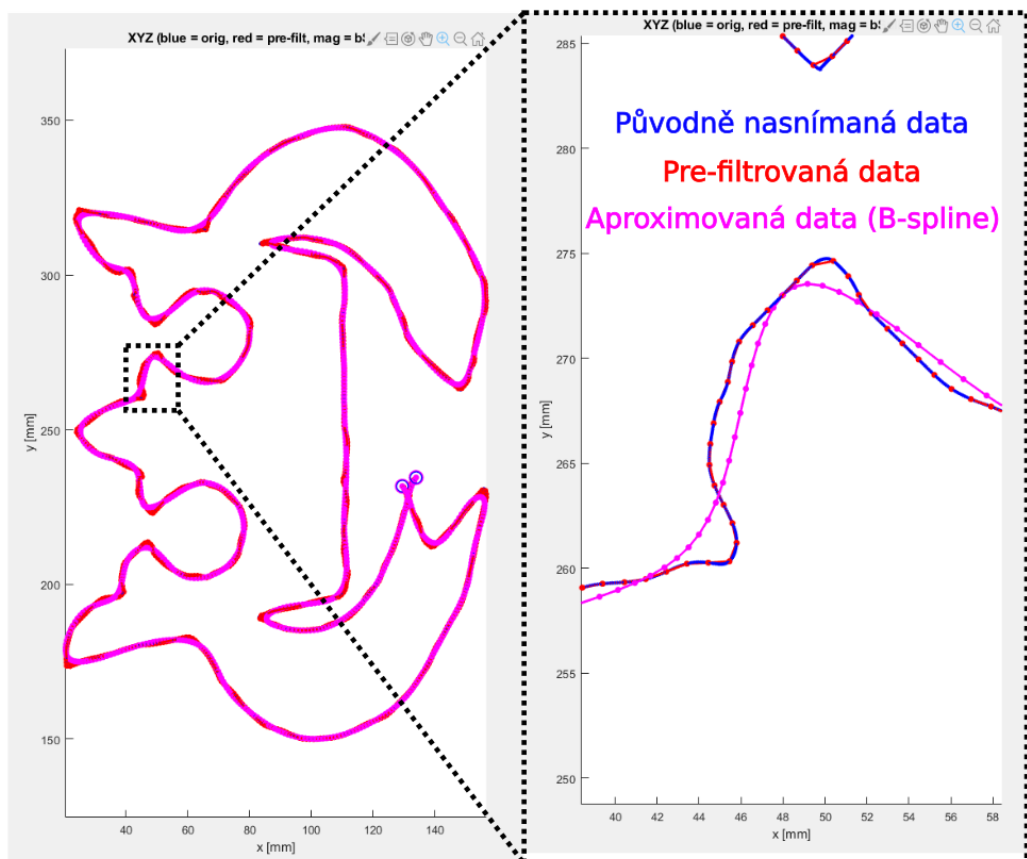


Obrázek 3: Časové průběhy XYZ polohy, rychlosti a profilu rychlosti zaznamenaných a filtrovaných dat

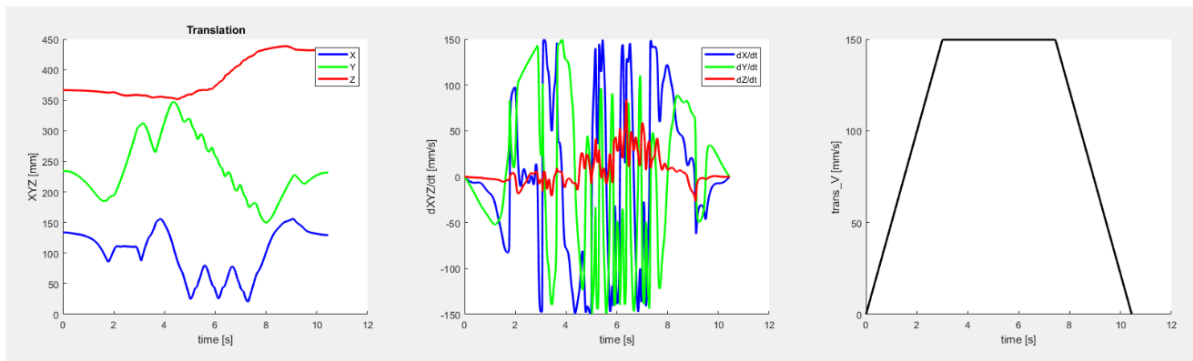
Aproximace naučeného pohybu s řízením profilu rychlosti

V některých aplikacích (např. nanášení lepidla, laserové kalení, atd.) je žádoucí, aby po naučení pohybové trajektorie robotu po této trajektorii robot udržoval definovaný profil rychlosti. Během procesu učení pohybu operátorem výroby však tento požadavek v podstatě nelze splnit (během učení dochází k fluktuaci rychlosti, zastavením na odpočinek, atd.). Pro tento případ byl implementován algoritmus aproximace naučené trajektorie s řízením profilu rychlosti. Obrázek 4 a 5 demonstruje filtraci polohového signálu zaznamenaného při učení pohybu robotu operátorem v prostoru XYZ a v časové oblasti (uvažována jen translace). Filtrace signálu použitým filtrem vykazuje následující vlastnosti:

- Pre-filtrace, která vybírá z původně naučených polohových bodů pouze body, které jsou od sebe vzdáleny více než zadaná hodnota v translaci a nebo orientaci => redukce počtu bodů vstupující do následné aproximace
- B-spline filtr s libovolně voleným řádem polynomu
- Možnost plynulého nastavení míry aproximace od striktní interpolace po vágní aproximaci
- Možnost definovat průběh rychlosti (v translaci nebo orientaci) podél trajektorie ve smyslu lichoběžníkového profilu rychlosti => optimální časový průběh rychlosti s omezením na maximální rychlost a zrychlení



Obrázek 4: XYZ poloha zaznamenaných, pre-filtrovaných a filtrovaných dat



Obrázek 5: Časové průběhy XYZ polohy, rychlosti a profilu rychlosti filtrovaných dat (konstantní profil rychlosti s omezením na max. rychlost a zrychlení)

Video demonstrace

Video ukázka z učení pohybu a následné filtrace výše uvedenými režimy je dostupná na odkazu:

<https://drive.google.com/file/d/1d1JMBGEEXcoV1qFf1A6G7K6gA6Vb6WvC/view?usp=sharing>

Reference

- [1] A. Jáger, A. Boublerle: *Analýza požadavků na pilotní aplikaci (systémová analýza)*, technická zpráva, IMI, 2022.
- [2] M. Švejda, A. Jáger, J. Reitinger: *Koncepce operátorské navádění robotu za účelem intuitivního programování*, technická zpráva, ZČU, 2022.