# Vývoj řídicího systému pro Laser Shock Peening stanici

# Úvod

## Zaměření mé výzkumné činnosti

## Laser shock peening process

Laser peening je mechanický proces pro zlepšení povrchových vlastností materiálu. Použitím laserového impulsu o vysoké energii je vytvořena rázová vlna, která se šíří materiálem. Účinkem rázové vlny se tvářením za studena zlepší mechanické vlastnosti povrchu materiálu. Parametry LSP procesu bývají následující:

* Nd:sklo laser na vlnové délce 1054 nm
* Délka impulsu 8 – 40 nm
* Energie impulsu 1 – 50 J
* Průměr svazku 1 – 6 mm
* Hustota výkonu 4 – 10 GW/cm2

Povrch materiálu je obvykle pokryt vrstvou barvy, fólií nebo páskou. Tato vrstva umožňuje absorbovat více energie než samotný povrch. Přídavná vrstva je laserovými impulsy zničena, ale povrch materiálu zůstává z velké míry nedotčen.

Laserový impuls při interakci s materiálem vypaří malou část povrchu materiálu. Vypařená vrstva dále pokračuje v absorbování energie a vytváří rychle se rozšiřující plazma. Vytvoření rázové vlny je důsledkem expanze plazmatu.

Celý proces laser peeningu musí probíhat pod udržující kapaliny, která zajišťuje, aby energie vytvořeného plazmatu nebyla uvolňována do okolí. Vrstva kapaliny usměrňuje rázové vlny dovnitř materiálu a proces se stává účinnějším. Kapalina musí být pro laserový svazek průhledná. Nejčastěji používanou kapalinou je voda.

## ROS: open-source Robot Operating System

ROS má sice v názvu slovní spojení operační systém, ale operačním systémem není. Jedná se o Framework, který má umožnit výzkumníkům v oblasti robotických systémů pomoct zvládnout spravovat softwarové projekty a urychlit vývoj softwaru pro roboty. Psát software pro robotické platformy je složitým úkolem – rozdílní roboti jsou vybaveni různým hardwarem, což značně omezuje opětovné využití kódu. Pro složitější aplikace může být pouhé množství kódu, které je třeba napsat být již odrazující. Je nutné programovat od nízkoúrovňových driverů po plánování trasy až po percepci. Tento záběr je nad časové schopnosti jednoho vývojáře, proto většinou na robotických projektech pracují celé týmy programátorů. ROS není nejlepším frameworkem pro software robotů, takový middleware nejspíš ani neexistuje – odvětví robotiky je na jedno řešení nejspíše moc rozsáhlé. Filozofie ROSu se dá shrnout do následujících bodů:

* Peer-to-peer
* Multi-jazykové
* Založený na distribuovaných nástrojích
* Open-source

Peer-to-peer

Systém postavený na ROSu se skládá z řady procesů, které obvykle běží na více hostech, které jsou spojeny peer-to-peer topologií. Peer-to-peer topologie vyžaduje, aby procesy byly shopny vzájemně se vyhledat za runtimu. V ROSu se tento mechanismus nazývá master nebo name service.

Multi-jazykové

ROS je vyvinut tak, aby byl jazykově neutrální. ROS v současné době podporuje čtyři velmi rozdílné jazyky: C++, Python, Octave a LISP. ROS pracuje jako message-oriented middleware (MOM). MOM je softwarová struktura, která podporuje odesílání nebo přijímání zpráv mezi distribuovanými systémy. MOM umožňuje aplikacím být distribuované přes heterogenní platformy a snižuje náročnost při vývoji aplikací, které jsou nasazeny na více operačních systémech a používají více síťových protokolů. Middleware vytvoří distribuovanou komunikační vrstvu, která abstrahuje vývojáře aplikací od detailů používaných operačních systémů a síťových rozhraní. Typickým příkladem služby poskytované MOM jsou API pro různé operační systémy. Pro usnadnění vývoje pro různé programovací jazyky nabízí ROS jazykově-neutrální Interface Definition Language (IDL) pro popis zpráv posílaný mezi dvěma moduly. IDL používá velmi krátké textové soubory, kterými popisuje pole jednotlivých zpráv. Generátory kódu pro ROSem podporované jazyky poté automaticky generují nativní implementace v daných jazycích.

Založený na distribuovaných nástrojích

ROS je založený na tzv. mikrokernelovém designu, kdy je k dispozici řada nástrojů pro kompilaci a běh komponent v ROSu nežli použití monolitického vývojového a běhového prostředí. Tyto nástroje vykonávají řadu úloh – navigace skrz zdrojové kódy, nastavení konfiguračních parametrů, měření využití šířky pásma, vykreslení dat, automatickou generaci dokumentace atd. . V ROSu je snaha přenést všechny služby do samostatných modulů.

Open-source

Celý zdrojový kód ROSu je veřejně přístupný. ROS je distribuován pod licencí BSD, která umožňuje vývoj jak komerčních tak nekomerčních projektů.

KONCEPCE ROSU

Mezi základní prvky architektury ROSu patří uzly, zprávy, služby a témata.

Uzly jsou procesy, které provádí výpočty. ROS je navržen modulárně – systém se obvykle skládá z mnoha uzlů. Termín uzel je ve smyslu ROSu volně zaměnitelný s termínem softwarový modul. Název uzel vychází ze způsobu vizualizace systémů za běhu: pokud v systému běží mnoho uzlů najednou, je vhodné zobrazit vztahy mezi nimi s pomocí grafu. Uzly spolu komunikují vzájemným posíláním zpráv. Zpráva je silně typovaná datová struktura. Jsou podporovány základní primitivní typy (integer, float, boolean) a také pole primitivních struktur a konstanty. Uzel posílá zprávu jejím zveřejněním na určité kategorii, která je určena jednoduše názvem jako například „odometrie“ nebo „mapa“. Uzel, který chce získat určitý typ dat, se poté přihlásí k odběru dané kategorie. Nejjednoduším typem je komunikace po pipeline, ale v realitě jsou grafy složené z uzlů o mnoho složitější, obsahují cykly, 1:N a M:N spojení.

I když systém přenášení zpráv založený na kategoriích je flexibilní, není vhodný pro synchronní komunikaci. V ROSu se místo toho používá takzvaných služeb. Služba je definovaná svým jménem a dvěma striktně typovanými zprávami – jednou pro požadavek a druhou pro odezvu.

Možnosti systému ROS a jeho výhody při použití pro vývoj robotických aplikací

Ladění uzlů

Při vývoji robotického systému v akademické sféře se obvykle zaměřujeme na vývoj specifického uzlu (procesu), který vykonává určitou úlohu v oblasti plánování trasy, strojového vidění nebo řízení. Pokud ale chceme uvést systém do provozu, potřebujeme pro to daleko větší již existující infrastrukturu. Musíme mít k dispozici fungující ovladače pro manipulátory, kamery a různé sensory, vizualizační software atd.. ROS s jeho modulární strukturou umožňuje vyvíjet a upravovat nové uzly bok po boku uzlů již existujících a ověřených. Dále graf uzlů a spojení mezi nimi upravovat za běhu, a tím graf dynamicky modifikovat. K již existující infrastruktuře můžeme tedy za běhu provádět změny na právě vyvíjeném novém uzlu.

Logování a následné přehrávání

Výzkum v oblasti robotiky je často prováděn se zalogovanými daty ze senzorů, což umožňuje srovnání různých algoritmů a zjednodušení experimentu. ROS umožňuje data logovat a následně znova přehrávat.

Zjednodušené spouštění více procesů

Spouštět jednotlivé uzly každého systému pokaždé zvlášť by bylo pro rozsáhlejší systémy obtížné. ROS nabízí možnost spouštět najednou více procesů. Popis systému je v tomto případě uložen v XML souboru a k jeho spouštění slouží utilita roslaunch.

Kolaborativní vývoj

Software v ROSu je organizován do balíčků, aby byl podpořen

Vizualizace a monitorování

Při vytváření a ladění softwaru v robotice se často setkáváme se situací, kdy je nutné zjistit stav systému za běhu. Pro velké distribuované systémy si poté už se jednoduchými metodami pro ladění nevystačíme. ROS a jeho architektura zpráv a služeb umožňuje se přihlásit k odběru libovolné kategorie a napsat uzel pro vizualizaci dat. Spolu s ROSem je distribuován program rviz pro vizualizaci dat, který umožňuje zobrazení dat ve formě trajektorií, renderovaných modelů robotů, obrazů nebo ve formě tzv. point cloudů.

? rostopic, rxplot

Prostory jmen v ROSu

Zásobníkem se v ROSu nazývá skupina uzlů, které vykonávají podobnou činnost. Někdy je potřebné inicializovat více instancí identického zásobníku. ROS umožňuje vytvoření samostatného prostoru jmen pro každou instanci zásobníku, kterou potřebujeme bez nutnosti následné modifikace zdrojového kódu.

Transformace

V robotice je důležité sledovat vztahy mezi jednotlivými vztažnými soustavami, ať už mezi robotem a okolím nebo robotem a vztažnými soustavami nejrůznějších sensorů. Pro ROS je k dispozici transformační systém s názvem tf, který uvádí jednotlivé soustavy do vztahů pomocí tzv. transformačního stromu. Ze získaných informací ze senzorů subsystémů (enkodéry kloubů), tf ulitita provede potřebné výpočty pro transformaci mezi vztažnými soustavami. ROS tímto způsobem značně zjednodušuje programátorovi práci, která je jinak velmi náchylná k chybám.

## ROS-Industrial

ROS-Industrial rozšiřuje možnosti použití softwaru ROS na průmyslové manipulátory. O budoucí vývoj ROS-Industrial se stará konsorcium ROS-Industrial, jehož evropská větev je řízena výzkumným ústavem Fraunhofer IPA v Německu. Hlavní cíle ROS-Industrial se dají shrnout do následujících bodů:\

* Zkombinovat ROS s již existujícími průmyslovými technologiemi a využít schopnosti ROSu pro použití v průmyslových procesech
* Vyvinout spolehlivý software pro průmyslové procesy
* Zjednodušit výzkum v oblasti průmyslové robotiky
* Vytvořit širokou komunitu vývojářů podporovanou výzkumníky a profesionály z průmyslové sféry

ROS-Industrial je nadstavbou ROSu, a tím pádem může těžit i ze všech jeho výhod. Mezi ty nejdůležitější patří:

* K dispozici jsou všechny pokročilé nástroje z ROSu – Rviz, Gazebo, nástroje pro inverzní kinematiku a další nástroje pro vizualizaci, simulaci a ladění.
* Aplikace připravené k okamžitému použití – ROS obsahuje aplikace pro strojové vnímání i aplikace pro uchopení i paletizaci a třídění komplexních objektů
* Nízká cena – na rozdíl od proprietárního robotického softwaru je ROS open source a je možné ho používat bezplatně
* Zjednodušení programování robotů – ROS odstraňuje problém s učením a plánováním trasy robotických manipulátorů a místo toho umožňuje automatické naplánování optimální trasy bez kolizí.

## ROS-ovladače pro manipulátory FANUC

Manipulátory FANUC jsou v ROSu podporovány sadou ovladačů pro kontroléry typu R-30iB, R-30iA a starší R-J3iC. Ovladače jsou dobře zdokumentované a doprovázeny nástrojem, který umožňuje jejich instalaci krok za krokem. Ovladače podporují streamování polohy, kdy jsou polohy kloubů streamovány kontroléru. Rychlost pohybu je pevně stanovena v kontroléru.

? psát o struktuře ovladačů pro roboty v ROS-I?

## Kontrola procesu Laser Shock Peeningu

Pro kontrolu kvality procesu LSP se předpokládá kombinace metod pro klasický peening (balotinování) pomocí Almenových pásků a elektronického monitorování procesu. Nejdříve je potřeba provést absolutní kalibraci energie laserového impulsu. Absolutní kalibrace bude provedena vložením kalorimetru do svazku před započetím procesu. Zároveň bude druhý kalorimetr vzorkovat malé procento energie impulsu (desetiny procenta celkové energie impulsu) a bude nakalibrován vzhledem k celkové energii svazku měřené prvním kalorimetrem. Druhý kalorimetr bude poté při běžícím procesu peeningu sloužit pro měření plné energie laserového impulsu. Dále bude pyrodetektorem měřena energie impulsu druhou nezávislou metodou. Tento pyrodetektor bude pomocí triggrovacího signálu připojen k časovacímu systému laseru. Uniformita svazku bude měřena CCD kamerou. Odezva kamery je lineární, a tudíž je jejím výstupem intenzita v profilu svazku.

Mezi pokročilé metody kontroly procesu LSP patří měření doby letu akustické rázové vlny vyvolané LSP procesem. V tomto případě je před vzorkem umístěn senzor, který měří čas vypálením laseru a detekcí akustické rázové vlny vyvolané LSP procesem na sensoru. Rychlost rázové vlny je úměrná energii uložené v materiálu při LSP procesu a lze z ní usuzovat na kvalitu provedení LSP procesu. Čím je detekovaná doba letu kratší, tím více energie bylo v materiálu uloženo, vlastnosti materiálu jsou tvářením zastudena zlepšeny do větší hloubky a proces je v konečném důsledku kvalitnější.

Na Obrázku je zobrazeno schéma celé aparatury.

Dále je potřeba získaná data dále zpracovat. Samozřejmostí je ukládání historie dat, které ROS podporuje. Je možné ukládat data z kamer i senzorů spolu s polohou robotického ramene v čase a následně je přehrávat a generovat z nich statistiky. Konečným cílem je řídit kvalitu procesu v reálném čase. V současně používaných LSP technologiích je nutné ověřovat kvalitu procesu rozsáhlým destruktivním zkoušením pro každou sérii dílů. To znamená, že ze zpracovaných dílů je několik náhodně vybráno a provedeny destruktivní zkoušky únavy materiálu. Destruktivní zkoušení je časově náročné, drahé a nedává informace o kvalitě LSP procesu na aktuálně zpracovávaném vzorku. Je proto žádoucí, aby byl vyvinut monitorovací systém, který umožňuje nedestruktivní monitorování kvality procesu v reálném čase. Na začátku by byly na malé sérií daných dílů určeny vhodné parametry LSP procesu (energie a uniformita impulsu, počet nutných opakování LSP procesu, množství udržovací kapaliny). Díly z této série jsou testovány - difrakcí rentgenového záření jsou zjišťovány průběhy napětí v materiálu indukované tvářením za studena, je provedeno únavové zkoušení dílů. Poté, co jsou stanoveny vhodné parametry procesu, jsou nastaveny limity

* v té