VŠB – Technická univerzita Ostrava

Fakulta elektrotechniky a informatiky

Katedra měřicí a řídicí techniky

Návrh, řízení a vizualizace třídičky barevných objektů s využitím robotické ruky L601-KT na platformě .NET Framework

2009 **Bc.Skovajsa Lukáš**

Klíčová slova: Robotická ruka, robotické manipulátory, Microsoft .NET Framework, Visual Studio, CodeWarrior, zpracování obrazu, kinematika, L601-KT, C#, HCS12, SSC32, RS232, PWM, Denavit-Hartenberg

Abstrakt: Tato práce řeší problematiku řízení robotické ruky s typovým označením L601-KT.

Pro ovládání robotické ruky je využíván zjednodušený kinematický model, díky kterému je možno ovládat robotickou ruku intuitivně v kartézském souřadném systému. Vytvořený software ovládá robotickou ruku na základě informací získaných z kamery a implementuje tak algoritmy zpracování a analýzy obrazu. Systém při své práci postupuje tak, že nejprve získá snímek z kamery, ten zpracuje a na základě informací z něj získaných vypočítává řídící příkazy pro jednotlivé servomotory. Nalezené předměty pak jeden po druhém třídí podle jejich barvy a velikosti do připravené šablony, která obsahuje odpovídající otvory lemované příslušnou barvou.

Aplikace je vytvořena v programovacím jazyce C# na platformě Microsoft .NET Framework. V rámci práce byl také vytvořen mechanický model s robotickou rukou, který disponuje řadou nastavitelných parametrů pro další využití ve výuce na Katedře měřicí a řídicí techniky.

1. Úvod

Automatizace, to je trend dnešní doby. Výrobní podniky stále více angažují ve své výrobě průmyslové manipulátory a roboty. Důvod tohoto kroků je zřejmý. Dochází ke zvýšení kvality práce, snížení nákladů, zvýšení výrobní kapacity, ale mnohdy je i důvodem zdraví a bezpečnost člověka.

V průmyslu bychom jistě našli spoustu aplikací, kde pro řízení robotických manipulátorů jsou využívány informace, získávány prostřednictvím kamerového subsystému. V tomto duchu bylo vytvořeno téma této diplomové práce, která si klade za úkol sestavit model, na kterém se studenti seznámí s možnostmi, jak lze řídit robotické manipulátory na základě informací získaných zpracováním obrazu.

Je postaven model, který obsahuje dvě kamery a robotickou ruku s typovým označením L601-KT. Současně je vytvořena demonstrační úloh, která využívá vytvořené algoritmy pro zpracování a analýzu obrazu, a také algoritmy pro řízení robotické ruky. Na základě těchto algoritmů pak třídí barevné objekty podle jejich velikosti a barvy.

2. Model třídičky barevných objektů

Podle papírového prototypu byl vytvořen model s hliníkového plechu. Model má několik povrchových úprav. První je vrstva tmelu, pro zakrytí velkých rýh a děr způsobených při opracovávání. Následují dvě vrstvy tmelu ve spreji, který slouží pro zakrytí menších nerovností a zároveň slouží jako základní barva. Další vrstvou je barva bílá, kterou jsou vytvořeny popisky na modelu. Předposlední vrstvou je barva černá, která tvoří konečnou barvu celé mechanické konstrukce. Nakonec je celý model opatřen dvěma vrstvami bezbarvého laku.

Obr. 1 Fotografie modelu třídičky barevných objektů

Jeden z hlavních požadavků na mechanickou konstrukci bylo, aby byl v maximální možné míře flexibilní. Nejdůležitější vlastností je bezpochyby možnost vyměnit pracovní plochu. Tato část pracovní plochy je k

modelu uchycena pomocí šestice magnetů, takže její demontáž a výměna je otázkou několika vteřin. Snadná je také možnost vyměnit elektronické moduly umístěné uvnitř modelu. Model je totiž opatřen dvěma DIN lištami, na které je možno vytvořené elektronické moduly umístit

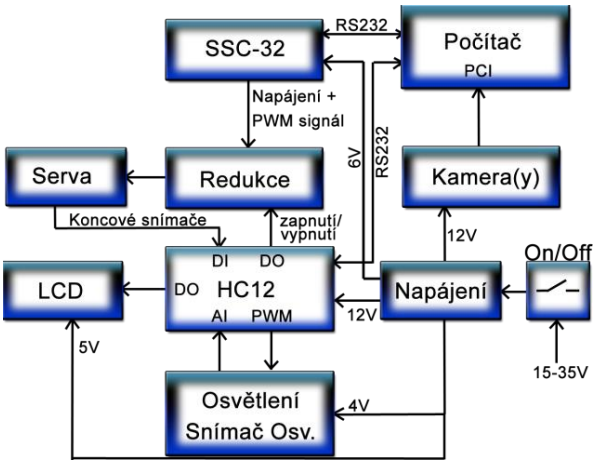
Pro model třídičky barevných objektů byly vytvořeny dvě šablony, které obsahují geometrické tvary různých barev a velikostí. Kostky jsou vyrobeny s balzy, natřeny příslušnou barvou a ze spodní strany jsou označeny číslem šablony, ke které patří.

Obr. 2 Detaily modelu třídičky barevných objektů

3. Blokové schéma modelu třídičky barevných objektů

Jádrem celého systému je řídicí počítač, ve kterém je realizován řídicí algoritmus, včetně vizualizace celého řídicího systému. Řídicí počítač zajišťuje zpracování signálu (z kamery, SSC-32, HCS12) a ovládání robotické ruky L601-KT.

Jednotka SSC-32 přijímá signály z řídicího počítače a podle zaslaných instrukcí generuje příslušný PWM signál pro všechny servomotory. Mezi jednotku SSC-32 a servomotory je zařazena redukce, která realizuje zapnutí popřípadě vypnutí servomotorů pomocí mikrokontroléru HCS12. Mikrokontrolér HCS12 na požadavek zjišťuje stavy koncových spínačů a tyto stavy zasílá řídicímu počítači. Intenzitu osvětlení je možno nastavit z řídicího počítače na konstantní hodnotu.



Obr. 3 Blokové schéma

4. Vytvořené a používané hardwarové zařízení

Blok Napájení má za úkol upravit velikost napájecího napětí na hodnotu 4V, 5V, 6V a 12V. První hodnota 4V slouží pro napájení osvětlení. Použity jsou dva 1W Luxeony. Každý z nich odebírá maximální proud 350mA. Hodnota 5V slouží pro napájení logiky modulu SSC-32, kde maximální odebíraný proud je do 250mA. Hodnota 6V slouží pro napájení servomotorů. Protože je použito šest servomotorů, přičemž jeden servomotor může mít při zatížení odběr až 500mA, je nutno tedy zajistit, aby byl tento zdroj schopen dodávat proud až 3A. Poslední stabilizované napětí je 12V. Toto napětí slouží pro napájení kamerového subsystému a mikrokontroléru HCS12.

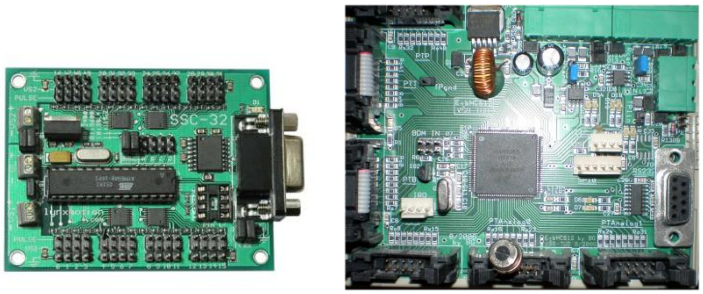
Blok Osvětlení slouží pro možnost ovládat přisvětlení a tím zajišťuje kvalitní obraz snímaný kamerou. Toto zařízení obsahuje snímač pro zjištění přibližné hodnoty osvětlení (aktuálně není využíván). Zařízení je připojeno k mikrokontroléru HCS12, který prostřednictvím PWM signálu, reguluje intenzitu osvětlení a udržuje tak její hodnotu na konstantní (nastavitelné) hodnotě

Blok Redukce umožňuje ovládat zapnutí/vypnutí servomotorů pomocí digitálního výstupu HCS12. Díky možnosti komunikace s řídicím PC jsou tyto digitální výstupy ovládány uživatelem

Obr. 4 Vytvořená hardwarová zařízení (Napájení, Osvětlení, Redukce)

Modul SSC-32 slouží pro řízení až 32 servomotorů s přesností 1μs (0.09°). Komunikace je zajištěna prostřednictvím sériového komunikačního rozhraní RS232. Z řídicího počítače lze zasílat pomocí RS232 řídicí signály, na základě kterých je pak vypočten PWM signál, který je nastaven na příslušný výstupní pin.

EvbHCS12 je modul určený pro výuku na Katedře měřicí a řídicí techniky. Obsahuje 16 bitový mikrokontrolér (MCU) s vnitřní 256KB Flash EEPROM pamětí, 12KB RAM a 4B EEPROM. Dále obsahuje dva osmi kanálové 10bitové A/D převodníky, 8 PWM kanálů (8 kanálů 8mi bitových, nebo 4 kanály 16-ti bitové), 29 digitálních I/O. Komunikace je umožněna prostřednictvím sériového rozhraní RS232.



Obr. 5 Použité hardwarové moduly

5. Kinematický model robotické ruky L601-KT

Přímá kinematika:

Úloha přímé kinematiky řeší případ, kdy známe jednotlivé kloubové souřadnice všech kloubů a chceme vypočítat polohu koncového efektoru. Tato úloha je pro otevřený kinematický řetězec snadno řešitelná pomocí geometrických vztahů mezi jednotlivými klouby.

Rozmístíme-li souřadné systémy podle Denavit-Hartenbergových principu, budou parametry pro robotickou ruku L601-KT:

Kloub	q_i	d_i	a_i	α_i
1	q_1	0	0	90
2	q_2	0	L_1	0
3	q_3	0	L_2	0
4	q_4	0	L_3	-90
5	q_5	0	0	0

Denavit-Hartenbergovy parametry stačí nyní dosadit do následující homogenní transformační matice:

$$A_{i-1,i} = \begin{bmatrix} \cos q_i & -\sin q_i \cos \alpha_i & \sin q_i \sin \alpha_i & a_i \cos q_i \\ \sin q_i & \cos q_i \cos \alpha_i & -\cos q_i \sin \alpha_i & a_i \sin q_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Zavedeme-li si označení: cosinus -> C a pro sinus -> S, tak transformační matice budou mít následující tvar:

$$A_{0,1} = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_{1,2} = \begin{bmatrix} C_2 & -S_2 & 0 & L_1C_2 \\ S_2 & C_2 & 0 & L_1S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_{2,3} = \begin{bmatrix} C_3 & -S_3 & 0 & L_2C_3 \\ S_3 & C_3 & 0 & L_2S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{3,4} = \begin{bmatrix} C_4 & 0 & -S_4 & L_3C_4 \\ S_4 & 0 & C_4 & L_3S_4 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_{4,5} = \begin{bmatrix} C_5 & -S_5 & 0 & 0 \\ S_5 & C_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

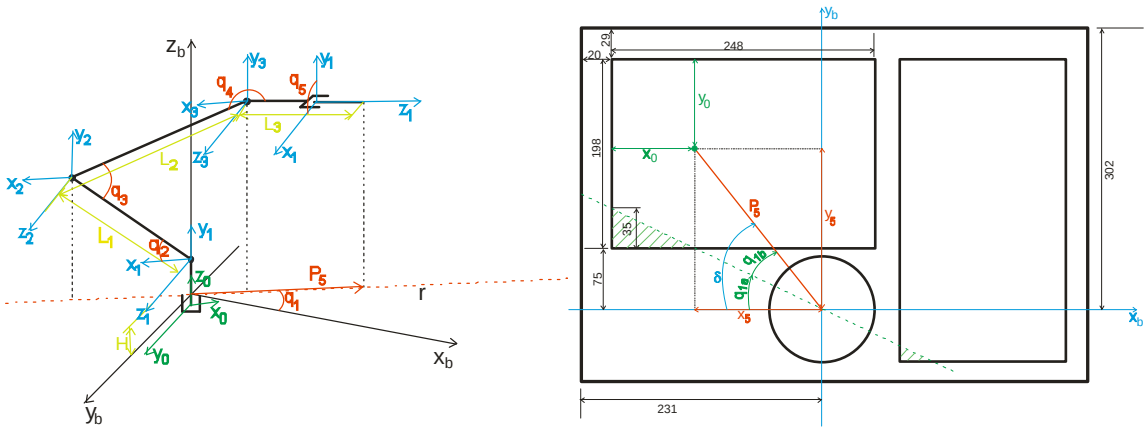
Celková transformační matice mezi základnou robotického ramene a efektorem je pak dána vynásobením těchto transformačních matic.

$$A_{0,5} = A_{0,1} \cdot A_{1,2} \cdot A_{2,3} \cdot A_{3,4} \cdot A_{4,5}$$

Inverzní kinematika

Inverzní kinematika řeší opačný problém, než kinematika přímá. Tato úloha je však mnohem složitější, než přímá úloha kinematiky, protože může mít více řešení a někdy dokonce i nekonečně mnoho řešení

Pro náš případ, však díky zavedenému omezení je možno zjednodušit řešenou úlohu z prostorové na rovinnou. Na obrázku 6 je zobrazen kinematický model robotické ruky L601-KT v prostoru. Jako GCS (Global Coordinate Systém) je označen kartézský prostor určen osami [x_b, y_b, z_b]. Robotický manipulátor operuje ve svisle orientované rovině *r*, jejíž natočení je dáno natočením osy prvního stupně volnosti (zobecněná souřadnice q_1). Prvním krokem je tedy určení natočení robotického manipulátoru o úhel q_1, dále pak můžeme brát mechanismus jako rovinný



Obr. 6 Kinematický model robotické ruky a výpočet souřadnice q_1

Úhel q_{1b} určíme takto:

$x_5 = 231 - (x_0 - 20)$; $y_5 = 302 - (y_0 - 29)$

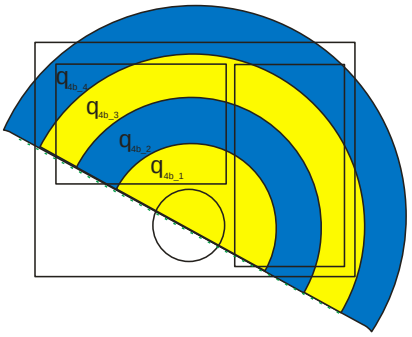
$$P_5 = \sqrt{x_5^2 + y_5^2}$$

$$tg(q_{1a}) = \frac{75 + 35}{231} \Rightarrow q_{1a} = arctg\left(\frac{110}{231}\right)$$

$$tg(\delta) = \frac{y_5}{x_5} \Rightarrow \delta = arctg\left(\frac{y_5}{x_5}\right)$$

$$q_{1b} = \delta - q_{1a}$$

Díky zjednodušení matematického modelu si zavedeme souřadnici q_4, jako parametr. Kdybychom nechali tento parametr konstantní, značně bychom tak snížili manipulační prostor robotické ruky. Proto tento parametr bude rozdělen podle toho, jak daleko se uchopovaný objekt od robotické ruky nachází (obrázek 7). Parametr q_{4b} je určován podle vzdáleností objektu, která je dána hodnotou P_5. Tato vzdálenost společně s hodnotami q_{4b_1}, q_{4b_2}, q_{4b_3} a q_{4b_4} byla určena experimentálně.



Obr. 7 Nastavení parametru q_{4b}

$$q_{4b_1} = q_{4b} = 134{,}1° \text{ pro } P_5 \leq 100 \text{ mm}$$
$$q_{4b_2} = q_{4b} = 120° \text{ pro } P_5 > 100 \text{ mm} \wedge P_5 \leq 190$$
$$q_{4b_3} = q_{4b} = 108° \text{ pro } P_5 > 190 \text{ mm} \wedge P_5 \leq 220$$
$$q_{4b_4} = q_{4b} = 100° \text{ pro } P_5 > 220 \text{ mm}$$

Výpočet inverzní kinematiky je nutno rozdělit ještě na dvě možnosti, které mohou nastat a to:

$$z_5 > H \text{ nebo } z_5 < H$$

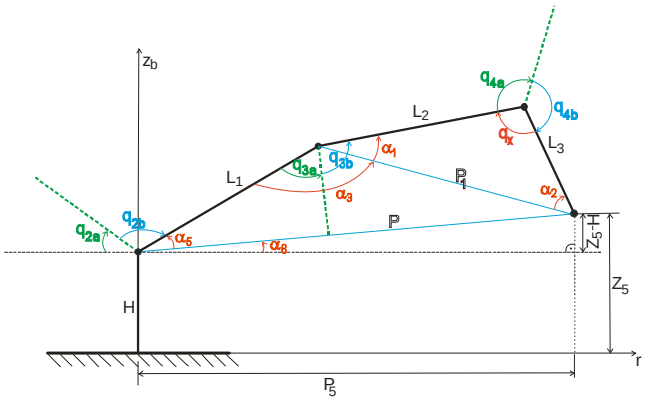
Kde hodnota z_5 udává výšku efektoru nad pracovní plochou modelu a hodnota H udává výšku základy, neboli posun souřadného systému LCS_1 vůči GCS. Na obrázku 8 je zobrazena první možnost a to $z_5 > H$. Výpočet natočení kloubů robotického manipulátoru provedeme, podle následujících vztahů

$$P_5 = \sqrt{x_5^2 + y_5^2};\ P = \sqrt{P_5^2 + (z_5 - H)^2};\ q_x = 360 - q_{4a} - q_{4b};\ P_1 = \sqrt{L_1^2 + L_3^2 - 2L_2L_3\cos\ q_x}$$

$$\alpha_1 = arcos\left(\frac{L_1^2 + P_1^2 - P^2}{2L_1P_1}\right) \quad \alpha_2 = arcos\left(\frac{L_3^2 + P_1^2 - L_2^2}{2L_3P_1}\right) \quad \alpha_3 = arcos\left(\frac{L_1^2 + P_1^2 - P^2}{2L_1P_1}\right)$$

$$\alpha_5 = arcos\left(\frac{L_1^2 + P^2 - P_1^2}{2L_1P}\right) \quad \alpha_6 = arsin\left(\frac{z_5 - H}{P}\right)$$

$$q_{3b} = (\alpha_3 + \alpha_1) - q_{3a};\ q_{2b} = 180 - \alpha_5 + \alpha_6 - q_{2a}$$



Obr 8 Schéma výpočtu inverzní kinematiky (pro z_5 > H)

Druhá možnost pro $z_5 < H$ je provedena analogicky.

Hodnoty q_{1a}, q_{2a}, q_{3a}, q_{4a} byly zjištěny měřením mechanických dorazů na robotickém manipulátoru. Tyto parametry zavádějí do výpočtu chybu způsobenou měřením.

Parametr	Jednotka	Hodnota
L1	mm	120,65
L2	mm	120,65
L3	mm	146,05
H	mm	76,20
q1a	°	25,46

q2a	°	43,92
q3a	°	58,41
q4a	°	123,57

6. Komunikace s mikrokontrolérem HCS12

Vytvořený algoritmus realizuje pět hlavních funkcí.

1. Příjem zpráv z řídícího počítače
2. Odesílání informací na požadavek řídicímu počítači
3. Nastavení intenzity osvětlení použitých Luxeonů
4. Čtení koncových snímačů
5. Zapínání/Vypínání jednotlivých servomotorů

Program funguje tak, že po zapnutí jsou všechny motory vypnuty, intenzita osvětlení je nastavena na nulu. V mikrokontroléru se nic neděje, dokud se na vstupu sériového komunikačního rozhraní neobjeví data. Při příjmu dat je vyvoláno přerušení, které se obslouží. Podle charakteru přijatých dat je vyhodnoceno, zda přišel požadavek na změření koncových snímačů a zaslání jejich stavu nebo zda přišli informace o tom, že se mají zapnout nebo vypnout motory popř. osvětlení.

7. Vytvořené metody a funkce pro zpracování obrazu a ovládání robotické ruky

Řízení robotické ruky

Námi využívaná je metoda inverzní kinematiky, která umožňuje ovládat robotickou ruku na základě souřadnic kartézského souřadného systému

```
/// Výpočet  inverzní kinematiky
/// </summary>
/// <param name="objekt">nalezené objekty v obraze</param>
/// <param name="index">číslo objektu, který chceme uchopit</param>
/// <param name="z0">výška efektoru</param>
/// <param name="celisti">velikost otevření čelistí v mm. rozsah 0-63 mm</param>
public void InverzniKinematika(List<FindObject.Objekt> objekt, int index, double z0, double celisti)
```

Pro potřeby ovládání robotické ruky byla vytvořena třída *RoboticArm*, která obsahuje kromě funkce inverzní kinematiky, také spoustu jiných funkcí. Ty nejdůležitější si nyní vyjmenujme.

```
// Zapne motory jeden po druhém s požadovanou časovou prodlevou
public bool ZapnoutMotoryPostupne(SerialPort serialPortHCS12, int cas)

// Zapne vybrané servomotory najednou
private void ZapnoutMotory(bool[] motory, SerialPort serialPortHCS12)

// Vrati maximalni(minimální) možnou pozici pro vybraný servomotor
public int VratHorniMez(int servo)
public int VratDolniMez(int servo)

// Nastavi servomotory do výchozí pozice
public void ResetServa()

// Nastaví servomotory do požadovaných úhlů
public void SetRoboticsArms(double Q1, double Q2, double Q3, double Q4, double Q5, double Q6)

// Roztřídí všechny objekty
public void RoztridObjekty(object tridObjekty)

// Roztřídí pouze jeden vybraný objekt
public void RoztridObjekt(object tridObjekt)
```

Po dokončení třídění předmětu je vyvolána událost *RoboticArmException*.
Využívání těchto funkcí je jednoduché. Stačí vytvořit novou instanci této třídy

```
RoboticArm L601KT = new RoboticArm(serialportSSC32);
```

A pak již využívat vytvořené funkce, například:

```
L601KT.ResetServa();
```

Vytvořená třída obsahuje i spoustu vlastností jako je *ArmLock*, *IsBusy*, *StopAction*, a jiné.

Zpracování a analýza obrazu

Také pro možnosti zpracování a analýzy obrazu byla vytvořena nová třída, která se jmenuje *FindObject*. I když tato třída implementuje spoustu dílčích funkcí, jsou vyžívány spíše interně a není potřeba je volat jednotlivě. Pro zpracování obrazu se provede následující příkaz:

```
// Nalezní všech objektů v obraze
List<FindObject.Objekt> objekty = FindObject.Process(snimany, (int)R, (int)G, (int)B, ZC);
```

kde funkce *Process*:

```
/// <summary>
/// Nalezne všechny objekty v obraze
/// </summary>
/// <param name="src">obrázek, ve kterém chceme objekty hledat</param>
/// <param name="r">R parametr filtru </param>
/// <param name="g">G parametr filtru</param>
/// <param name="b">B parametr filtru</param>
/// <param name="ZC">true-oblast předmětů, false-oblast šablon</param>
/// <returns>vrací kolekci nalezených objektů</returns>
public static List<Objekt> Process(Bitmap src, int r, int g, int b, bool ZC)
```

Přes proměnnou *objekty*, tak získáme přístup ke všem nalezeným předmětům v obraze a ke všem jeho zjišťovaným vlastnostem. Nyní si ve stručnosti vysvětlíme, jak třída *FindObject* funguje.

```
public static List<Objekt> Process(Bitmap src, int r, int g, int b, bool ZC)
{
        Bitmap bmp = new Bitmap(src);              // Vytvoření kopie Bitmapy
        var areas = Areas(bmp, r, g, b);           // Nalezení oblasti
        if (areas == null) return null;            // Bitmapa má špatny formát
        var allpnt = EdgePoints(areas);            // Souřadnice hranic objektů
        if (allpnt == null) return null;           // Nenalezeny žádné hranice
        var poly = EdgeLines(allpnt, bmp);         // Roztřídění bodu na polygony
        poly = ModifyPolygon(poly);                // Vrací polygony umístěné v pracovní oblasti
        var objekt = Identification(bmp, poly, ZC); // Provede identifikaci všech polygonů
        return objekt;                             // Vrací nalezené objekty
}
```

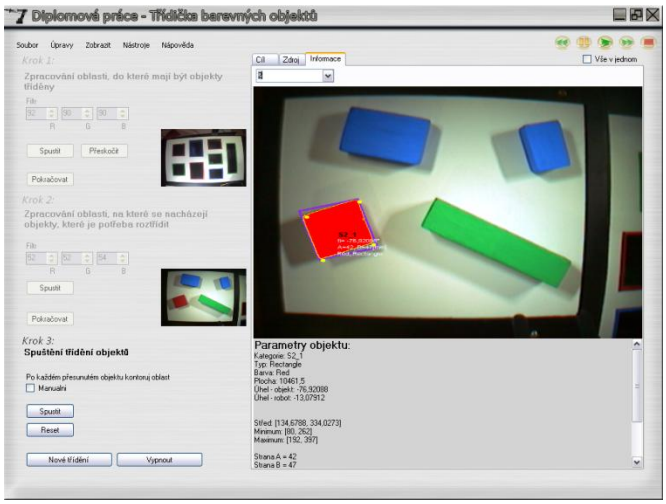
Na nalezených objektech (polygonech) je provedena identifikace a jsou zjištěny vlastnosti těchto polygonů. Jmenujme si alespoň některé:

- Centre: Souřadnice středu polygonu
- Color: Barva objektu
- Angle: úhel natočení objektu v obraze
- AngleForArms: úhel natočení, pod kterým musí být předmět uchopen robotickou rukou

Mezi další zjišťované vlastnosti patří kategorie, souřadnice rohů, plocha objektu, a další

8. Hlavní aplikace

Hlavní ovládací rozhraní pro model třídičky barevných objektů spustíme volbou *Soubor/Spustit*. Aplikace uživatele intuitivně vede k akcím, které má provést, aby bylo třídění úspěšné. Celý proces se skládá ze tří kroků. Ten první umožní snímat obraz z kamery nad oblastí, kde je umístěna šablona. Tento krok je však možné přeskočit, jelikož výsledky tohoto zpracování nejsou přímo využity v algoritmech pro řízení robotické ruky. Druhý krok již přeskočit nelze, jelikož v tomto kroku probíhá snímání a zpracovávání obrazu z oblasti, kde se nacházejí předměty potřebné k roztřídění. Uživatel může zkontrolovat, zda náhodou nedošlo ke špatnému zpracování a identifikaci obrazu. V případě, že je vše v pořádku kliknutím na tlačítko pokračovat, se zobrazí ovládací prvky kroku číslo 3.

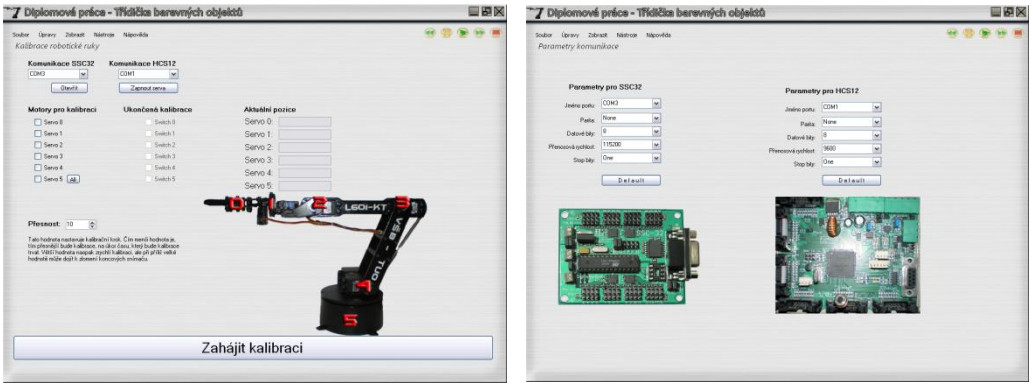


Obr. 9 Třídění barevných objektů

V tomto kroku je možno si vybrat, zda chceme manuální nebo automatický režim. Při výběru manuálního režimu si uživatel vybírá předmět, který chce roztřídit. Tento výběr se provede v menu, které je po volbě manuálního režimu zobrazeno. Automatický režim pak třídí všechny objekty, které se v pracovní ploše nacházejí a to zleva doprava. Robotickou ruku je možno v jakémkoli stádiu práce přerušit tlačítkem reset a nastavit jí tak do výchozí pozice. Při probíhajícím třídění je současně aktualizováno uživatelské rozhraní a objekt, který je roztříděn má vykreslen pouze fialový obrys a zvýrazněné rohy. Tím uživatel jednoznačně rozliší, které předměty se již na pracovní ploše nenacházejí.

Další možnosti je vypnout motory nebo začít nové třídění, čímž se dostáváme zpět do kroku 1.

Uživatel může pomocí grafického rozhraní změnit rychle základní parametry komunikace (obr. 10 vpravo). Krom výběru použitých portů bych však doporučoval tyto parametry neměnit, jelikož tyto parametry musí korespondovat i s nastavením hardwarových jednotek



Obr. 10 Kalibrace robotické ruky a parametry komunikace

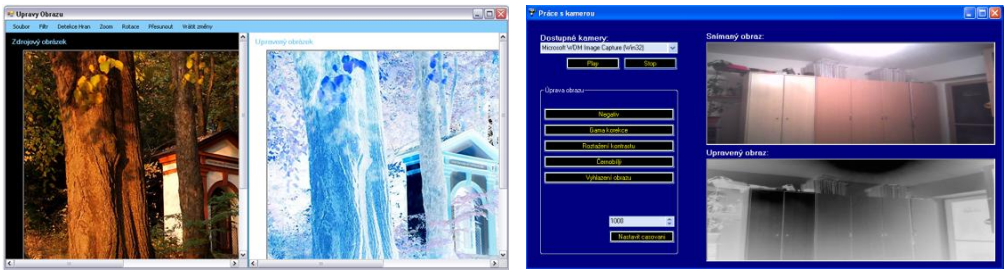
Kalibrace (obr. 10 vlevo) robotické ruky a koncové snímače byly realizovány z důvodu zabezpečení správné funkčnosti robotické ruky. Při dlouhodobém zatížení se použité servomotory zahřívají a díky tomuto vlivu se mění poměry v regulačním obvodu, který nastavuje natočení servomotoru na požadovanou polohu. Může se tak stát, že při dlouhodobé zátěži bude při stejné střídě vstupního signálu nastaven servomotor na jinou hodnotu úhlu natočení, než tomu bylo při zapnutí aplikace. Protože koncové snímače jsou pevně připevněny k mechanické konstrukci, je zjišťována jaká hodnota střídy vstupního signálu odpovídá natočení servomotoru do příslušné polohy. Díky tomu je zjišťován offset, který při oteplování servomotoru vzniká.

Uživateli je možno v rámci aplikace ovládat robotickou ruku pomocí joysticku, klávesnice nebo myši. Další důležitou vlastností je možnost si zobrazit tzv. Monitor. V tomto monitoru je zobrazena veškerá komunikace, která proběhla mezi řídicím počítačem a hardwarem. Aplikace také disponuje vlastní nápovědou.

9. Programy určené pro výuku

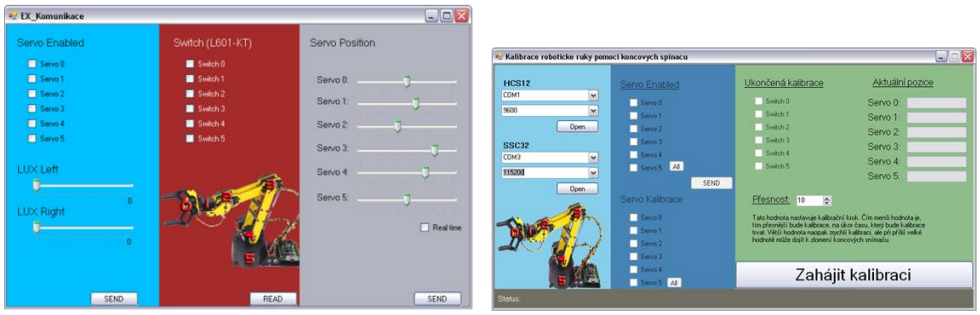
Z důvodů jednoduché srozumitelnosti práce s vytvořeným modelem a s vytvořenými algoritmy, byly jednotlivé problémy, které byly v rámci této diplomové práce řešeny dekomponovány.

Úprava obrazu je program, který disponuje celou řadou možných algoritmů, které lze na obraz aplikovat. Byla vytvořena samostatná knihovna, ze které jsou jen volány funkce s příslušnými parametry. Tento program tak slouží jako jeden z prvních kroků k pochopení, jak s obrazem v programovacím jazyce C# pracovat.
Práce s kamerou slouží pro ukázku toho, jak lze pracovat s kamerou v programovacím jazyce C# na platformě Microsoft .NET Framework. Program také obsahuje možnosti provádět korekce na snímaném obraze (např. negativ, gama korekce, atd.).



Obr. 11 Úprava obrazu (vlevo) a práce s kamerou (vpravo)

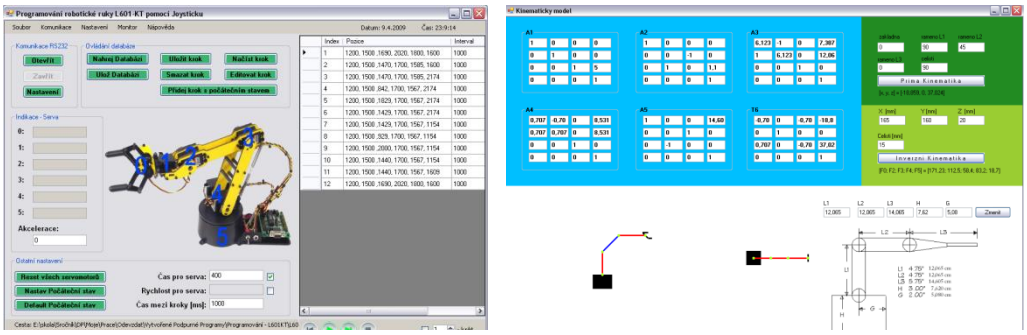
EX_Komunikace slouží pro jednoduchou ukázku komunikace s HCS12 a SSC32 pomocí sériového komunikačního rozhraní RS232. EX_Kalibrace slouží pro ukázku kalibrace robotické ruky pomocí koncových snímačů.



Obr. 12 EX_Komunikace (vlevo) a EX_Kalibrace (vpravo)

Aplikace Programování robotické ruky L601-KT pomocí joysticku umožňuje ovládání robotické ruky pomocí joysticku Logitech Force 3D PRO Pomocí joysticku lze naprogramovat nepřeberné množství kreací a různých pohybů, které lze mít uloženy v databázi. Tuto databázi lze v tomto programu zároveň editovat, upravovat, mazat nebo přidávat záznamy do kteréhokoli místa již vytvořené databáze. Tato aplikace je obsáhlejší, a proto má vlastní nápovědu, která vysvětluje všechny funkce a možnosti tohoto programu.

Aplikace s názvem Kinematický model ukazuje implementaci jak přímé tak inverzní kinematiky. Pro samotnou diplomovou práci nebylo potřeba vytvářet algoritmy pro metody přímé kinematiky. Metoda přímé kinematiky v tomto programu nerespektuje mechanická omezení robotického manipulátoru a představuje pouze obecné řešení. Naproti tomu inverzní kinematika byla vytvořena na míru této diplomové práci.



Obr. 13 Programování robotické ruky (vlevo) a Kinematický model (vpravo)

V rámci diplomové práce byly vytvořeny i jiné programy, které jsou popsány v příloze k diplomové práci.

10. Závěr

V rámci této práce byla vytvořená aplikace, která má za úkol třídit barevné objekty podle velikosti a barvy. Vytvořený model obsahuje dvě kamery, díky kterým systém získává informace ze svého okolí. Pro třídění objektů je využívána hlavně jedna kamera, která míří na prostor, kde jsou uloženy objekty, které chceme roztřídit. Díky vytvořeným algoritmům pro analýzu a zpracování obrazu detekuje řídicí systém, kde se objekty nacházejí, jak jsou natočeny, jakou mají velikost a barvu. Další zjišťovanou vlastností je, kde se nachází protikus tohoto objektu v dané šabloně. Po zjištění těchto informací čeká systém na příkaz obsluhy. Obsluha si může vybrat, zda chce roztřídit všechny objekty nebo jen některé.

Cílem diplomové práce je realizovat výukový model s robotickou rukou L601-KT a tím přispět ke zkvalitnění výuky na Katedře měřicí a řídicí techniky. Mechanický model za tímto účelem vytvořený, bere na vědomí nepřeberné množství aplikací, které mohou na základě této práce vzniknout a je ve všech svých směrech dimenzován tak, aby tento růst umožnil. Možnost měnit pracovní plochu modelu, tak může dát vzniknout každý rok zcela nové aplikaci v rámci studentských projektů.

Možností jak rozšířit nebo vylepšit tuto aplikaci je celá řada. V modelu by mohlo být například upraveno hardwarové řešení, které by zjednodušilo aplikaci dvou komunikačních rozhraní na použití pouze jednoho. Kinematický model robotického manipulátoru by mohl být implementován přímo do mikrokontroléru, včetně funkcí pro vizualizaci na LCD. Tohle by v mnohém usnadnilo práci s tímto modelem. Také algoritmy pro zpracování obrazu je možno dále vylepšovat. Například o možnost adaptovat se při měnících se světelných podmínkách nebo o možnost detekovat různé geometrické tvary. Velice zajímavou možností by pak byla 3D vizualizace robotické ruky například pomocí standartu OpenGL.

Literatura

[1] PISKAČ, L. *Průmyslové roboty*, Plzeň, 1999, ISBN 80-7082-554-5
[2] DOBEŠ, M. *Zpracování obrazu a algoritmy v C#*, Praha, 2008, ISBN 978-80-7300-233-6
[3] HLAVÁČ, V. *Zpracování signálů a obrazů*, Praha, 2002, ISBN 80-01-02114-9
[4] SOJKA, E. *Digitální zpracování a analýza obrazů*, Ostrava, 2000, ISBN 80-8078-746-5
[5] KARGER, A. *Základy robotiky a prostorové kinematiky*, Praha 2000, ISBN 80-01-02183-1
[6] GREPL, R. *Kinematika a dynamika mechatronických systémů*, Brno 2007, ISBN 978-80-214-3530-8
[7] HLAVÁČ, V. *Počítačové vidění*, Praha, 1992, ISBN 80-85424-67-3
[8] ŠOLC, F. *Robotika*, VUT Brno, 2002, skripta
[9] KOTEK, Z. *Metody rozpoznávání a jejich aplikace*, Praha, 1993, ISBN 80-200-0297-9
[10] KOT, T. *Mechatronický model robotu OJ10 v prostředí MCS/ADAMS*, diplomová práce, VŠB- TU Ostrava, 2004