

**HOCHSCHULE REUTLINGEN**  
**Reutlingen University**  
**– Studiengang Mechatronik Master –**

**Projektdokumentation**

**Image Understanding**  
**„Objektverfolgung – AUTOMATIC SOCCER TABLE“**

Gruppe 6  
Lauf, Christian  
Scheu, Tobias  
Weber, Thomas



## **Vorwort & Danksagung**

Eine Projektdokumentation ist das wissenschaftliche Abschlusswerk eines jeden Zweitsemester-Masterstudenten, und sie ist gekennzeichnet durch eine Phase unzähliger thematischer Diskussionen, umfangreicher Überarbeitungen und manchmal auch der schlaflosen Nächte.

Diese spannende Phase unseres Studiums liegt nun hinter uns, und es ist an der Zeit, uns bei allen zu bedanken, die uns in den vergangenen Monaten so tatkräftig unterstützt haben. Ohne ihre unendliche Geduld und Diskussionsbereitschaft wäre diese Dokumentation nicht zu dem Ergebnis geworden, das jetzt vor Ihnen liegt.

Unseren Professoren und Projektbetreuern sind wir für ihre unerschöpfliche Vielfalt an thematischen Ratschlägen und zielführenden Hinweisen auf dem Weg zum fertigen Projekt dankbar, doch auch unserem Kommilitonen Daniel Pohlmeier gebührt unser Dank für unzählige thematische Diskussionen, die diesen Projekten einen zusätzlichen Tiefgang und eine respektable thematische Vielfalt brachten.

## Erklärung

Wir versichere, dass wir diese Projektdokumentation selbstständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie alle wörtlich und sinngemäß übernommenen Stellen in der Arbeit gekennzeichnet haben. Die Projektdokumentation wurde noch keiner Kommission zur Prüfung vorgelegt und verletzt in keiner Weise Rechte Dritter.

Reutlingen, Datum

---

Unterschrift

# Inhaltsverzeichnis

Vorwort & Danksagung .....	II
Erklärung .....	III
Inhaltsverzeichnis .....	IV
Abbildungsverzeichnis .....	1
Abkürzungsverzeichnis .....	2
1 Abstract .....	3
2 Praktische Erweiterungen .....	4
2.1 Abdeckstutzen .....	4
2.2 PE-Anschluss .....	6
3 Theoretischer Bestandteil .....	7
3.1 Aktueller Algorithmus .....	7
3.2 C-Spline-Algorithmus .....	8
3.3 Vektorfeld .....	10
4 Ausblick .....	12
5 Anhang .....	12
5.1 Quellcode Matlab® .....	I
6 Literaturverzeichnis .....	13

## Abbildungsverzeichnis

Abbildung 1.1: Blockschaltbild Tischkicker .....	3
Abbildung 2.1: Provisorische Abdeckung .....	4
Abbildung 2.2: Bohrungen .....	4
Abbildung 2.3: Verschraubung .....	4
Abbildung 2.4: Abdeckstutzen .....	5
Abbildung 2.5: Zubehör .....	5
Abbildung 2.6: PE-Leiter abgetrennt.....	6
Abbildung 2.7: Adern abisoliert .....	6
Abbildung 2.8: Aderendhülsen aufgecrimpt .....	6
Abbildung 2.9: PE-Leiter angeschlossen.....	6
Abbildung 3.1: Vergleich der Algorithmen.....	10
Abbildung 3.2: Balltrajektorie in Vektorfeld .....	11

## **Abkürzungsverzeichnis**

BV Bildverarbeitung

GUI Graphical User Interface

MC Motion Control

# 1 Abstract

Der sogenannte automatische Tischkicker besteht aus den Komponenten BV und MC, schematisch in Abbildung 1.1 dargestellt. Das BV-System dient der Erfassung der Ballpositionen und berechnet die Torwartachssollposition. Das MC-System positioniert die Torwartachse.

Diese Dokumentation zeigt die Ergebnisse der Projektarbeit „Objektverfolgung – AUTOMATIC SOCCER TABLE“ im Rahmen der Veranstaltung Image Understanding des Masterstudienganges Mechatronik an der Hochschule Reutlingen.

Das Projekt und die Vorarbeiten beruhen auf den Bachelorthesen „Konzeption und Realisierung eines automatischen Torwarts für ein Tischfußballspiel – Teil 1: Bildverarbeitung“<sup>1</sup> und „Konzeption und Realisierung eines automatischen Torwarts für ein Tischfußballspiel – Teil 2: Motion Control“<sup>2</sup>.

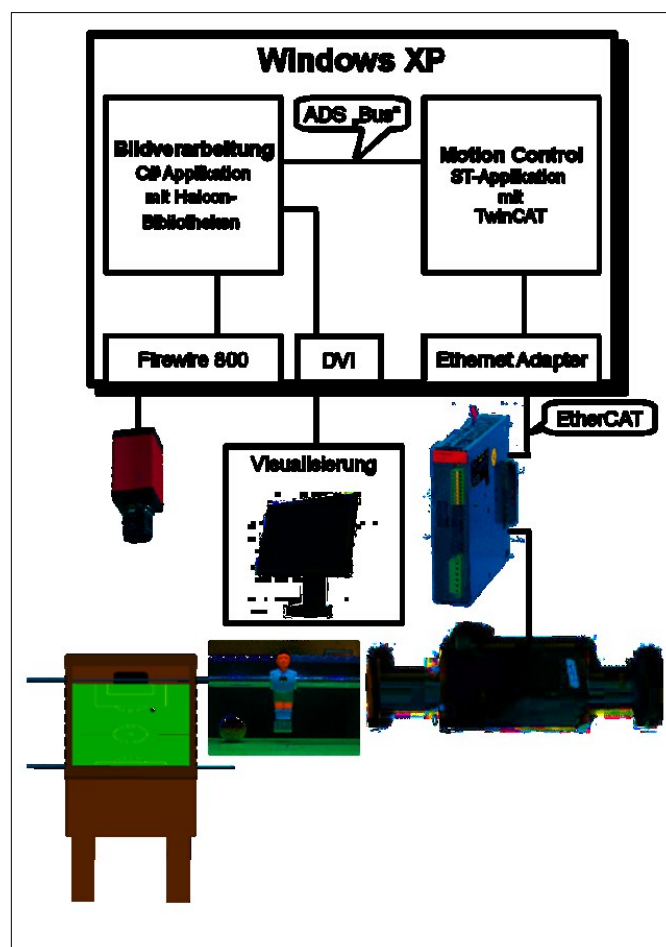


Abbildung 1.1: Blockschaltbild Tischkicker

## 2 Praktische Erweiterungen

Dieses Kapitel dient der Darstellung der praktischen Erweiterungen im Rahmen dieser Projektarbeit. Zunächst wird in Kapitel 2.1 die Ersetzung einer provisorischen Vierkantaluminiumprofilabdeckung durch Installation eines ordnungsgemäßen, sicheren Abdeckstutzens für die Torwartachse beschrieben. Anschließend wird in Kapitel 2.2 eine lebensgefährliche Elektroinstallation behoben, um Konformität mit DIN VDE 100 herzustellen.

### 2.1 Abdeckstutzen

Abbildung 2.1 zeigt die provisorische Abdeckung. In Abbildung 2.2 sind die Durchgangsbohrungen und Vorbohrungen für M10 Schrauben zu sehen. Folgend ist in Abbildung 2.3 dargestellt, wie der Abdeckstutzen mit Zylinderschrauben M10x50mm mit Innensechskant nach DIN 912 und Unterlegscheiben für M10 Form A nach DIN 125 A verschraubt ist.



Abbildung 2.1: Provisorische Abdeckung



Abbildung 2.2: Bohrungen

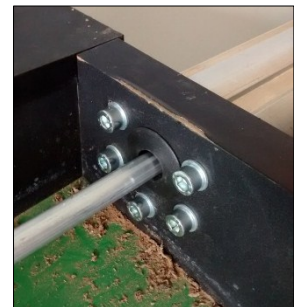


Abbildung 2.3: Verschraubung

(verbleibende Seite absichtlich leer gelassen)

Der neue Abdeckstutzen umschließt die Torwartachse auch bei maximaler Ausfahrlänge und verhindert so gefährliche Verletzungen, zu erkennen in Abbildung 2.4. Zusätzlich werden Ersatzteile in einem Kunststoffbeutel, angebracht am Abdeckstutzen – in Abbildung 2.5 verdeckt durch den Hinweiszettel –, beigelegt.



Abbildung 2.4: Abdeckstutzen



Abbildung 2.5: Zubehör

(verbleibende Seite absichtlich leer gelassen)

## 2.2 PE-Anschluss

Bei einer Elektroinstallation ohne angeschlossenen PE-Leiter liegt im Fehlerfall die volle Versorgungsspannung von  $U = 230V$  an den Metallprofilen an, siehe Abbildung 2.6. Durch einen vorschriftsmäßigen Anschluss des PE-Leiters kann diese Lebensgefahr gebannt werden. Dazu werden die Adern zuerst abisoliert, zu sehen in Abbildung 2.7. Anschließend werden Adererdhülsen  $1\text{mm}^2$  aufgecrimpt, dargestellt in Abbildung 2.8. Schließlich wird die Netzleitung wieder angeschlossen, siehe Abbildung 2.9.

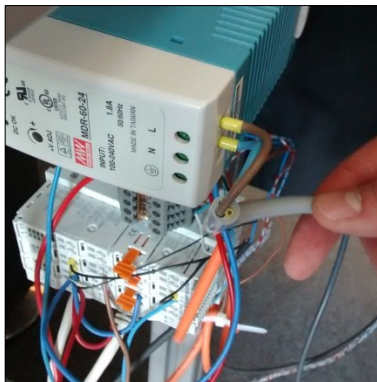


Abbildung 2.6: PE-Leiter abgetrennt

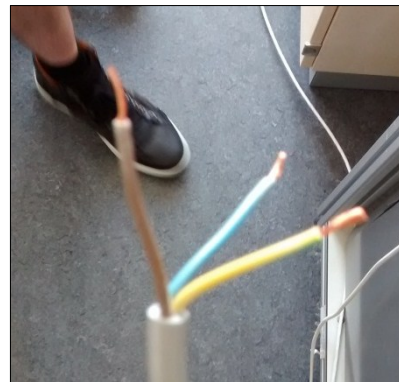


Abbildung 2.7: Adern abisoliert

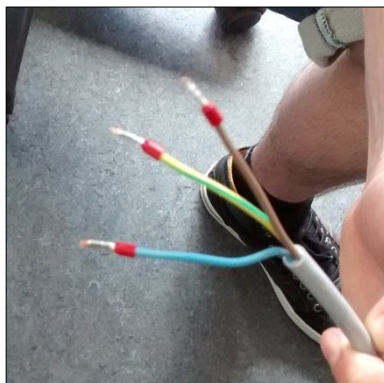


Abbildung 2.8: Aderendhülsen aufgecrimpt

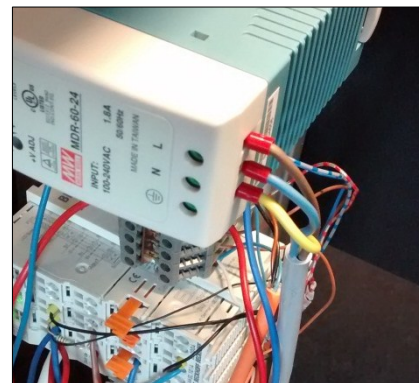


Abbildung 2.9: PE-Leiter angeschlossen

### 3 Theoretischer Bestandteil

In Kapitel 3.1 wird der zurzeit verwendete Algorithmus zur Ermittlung der Balltrajektorie, die zur Ermittlung der Torwartachssollposition verwendet wird, kurz beschrieben. Ein verbesserter, realitätsnaher Algorithmus wird in Kapitel 3.2 vorgestellt. Abschließend wird in Kapitel 3.3 ein idealer realitätsgetreuer Algorithmus aufgezeigt.

#### 3.1 Aktueller Algorithmus

Der aktuelle Algorithmus approximiert die Balltrajektorie mit einer Geraden. Die Anzahl der Ballpositionen, die zur Berechnung der Geraden verwendet werden, kann in der GUI eingestellt werden. Der verwendete Berechnungsalgorithmus ist abhängig von der Anzahl der eingestellten Ballpositionen.

Für  $n = 2$  Ballpositionen wird die Gerade mit Hilfe der Zweipunkteformel (3.1) berechnet.

$$y_{Torwart-Soll} = y_1 + \frac{y_2 - y_1}{x_2 - x_1} (x - x_1) \quad (3.1)$$

Für  $n = 3$  Ballpositionen werden zunächst zwei Geraden durch die Ballpositionen  $P_1$ ,  $P_2$  und  $P_2$ ,  $P_3$  ebenfalls nach (3.1) berechnet. Es ergeben sich zwei Torwartachsenpositionen  $y_1$ ,  $y_2$ . Die Torwartachssollposition  $\bar{y}_{Torwart-Soll}$  ergibt sich dann aus dem Mittelwert der beiden Torwartachsenpositionen nach Formel (3.2).

$$\bar{y}_{Torwart-Soll} = \frac{1}{n-1} \sum_{i=1}^{n-1} y_i \quad (3.2)$$

Für  $n > 3$  Ballpositionen wird eine Fehlererkennung und -korrektur durchgeführt. Es wird erst nach (3.1) und (3.2) die Torwartachssollposition  $\bar{y}_{Torwart-Soll}$  berechnet. Anschließend wird die Standardabweichung  $s$  der einzelnen Torwartachsenposition  $y_i$  zur Torwartachssollposition  $\bar{y}_{Torwart-Soll}$  nach (3.3) berechnet.

$$s = \sqrt{\frac{1}{n-2} \sum_{i=1}^{n-1} (y_i - \bar{y}_{Torwart-Soll})^2} \quad (3.3)$$

Torwartachsenposition  $y_i$  die mehr als  $s + 3$  abweichen, werden nicht für die endgültige Berechnung der Torwartachssollposition  $\bar{y}_{Torwart-Soll}$ , die wieder nach (3.2) berechnet wird, verwendet.

### 3.2 C-Spline-Algorithmus

Wie in Kapitel 3.1 gezeigt wird die Balltrajektorie lediglich mit einer Geraden approximiert – eine lineare Regression. Dies ist für sehr schnell geschossene Bälle eine akzeptable Lösung. Bei langsam geschossenen Bällen wirkt sich jedoch der Drall (und andere physikalische Gegebenheiten) stark auf die Balltrajektorie aus – sie ist keine Gerade mehr, sondern eine Kurve. Diese realitätsnähere Balltrajektorie wird mit Hilfe von C-Splines berechnet. Nachfolgend ist der Berechnungsalgorithmus dargestellt.

Gegeben: Ballpositionen  $P_i(x_i, y_i)$  für  $i \in [0, m]$

Gesucht: Parametrische Darstellung

Parametervektor  $u = \{u_i\}$ ,  $u_0 < u_1 < \dots < u_m$

Für jedes Intervall  $[u_i, u_{i+1}]$  für  $i \in [0, m - 1]$

$$f_{xi}(u) = a_{xi} + b_{xi}(u - u_i) + c_{xi}(u - u_i)^2 + d_{xi}(u - u_i)^3$$

$$f_{yi}(u) = a_{yi} + b_{yi}(u - u_i) + c_{yi}(u - u_i)^2 + d_{yi}(u - u_i)^3$$

Zunächst wird der Parametervektor  $u$  in Abhängigkeit des Abstandes der Ballpositionen  $P_i$  (Sehnenlänge) berechnet.

Berechnung der Sehnenlängen  $h_i$ :

$$h_0 = 0$$

$$h_i = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \text{ für } i \in [1, m]$$

Berechnung der Parameterwerte  $u_i$ :

$$u_i = \sum_{j=0}^i h_j$$

Die Berechnung der Polynomkoeffizienten  $a_{xi}, b_{xi}, c_{xi}, d_{xi}, a_{yi}, b_{yi}, c_{yi}$  und  $d_{yi}$  erfolgt anhand folgender Bedingungen.

Berührstetigkeit:

$$\begin{aligned}x_i &= f_{xi}(u_i) \\y_i &= f_{yi}(u_i) \\x_{i+1} &= f_{xi}(u_{i+1}) \\y_{i+1} &= f_{yi}(u_{i+1})\end{aligned}$$

Tangentiale Stetigkeit:

$$\begin{aligned}\frac{df_{xi}(u_i)}{du} &= \frac{df_{xi+1}(u_i)}{du} \\ \frac{df_{yi}(u_i)}{du} &= \frac{df_{yi+1}(u_i)}{du}\end{aligned}$$

Höhere Stetigkeit:

$$\begin{aligned}\frac{d^2 f_{xi}(u_i)}{du^2} &= \frac{d^2 f_{xi+1}(u_i)}{du^2} \\ \frac{d^2 f_{yi}(u_i)}{du^2} &= \frac{d^2 f_{yi+1}(u_i)}{du^2}\end{aligned}$$

Dadurch ergibt sich ein Gleichungssystem mit  $4m - 2$  Gleichungen und  $4m$  Unbekannten für  $x$  und  $y$ , wodurch sich jeweils zwei Freiheitsgrade ergeben. Dadurch kann z.B. die Steigung und Krümmung im Kurvenausgangspunkt gewählt werden.

$$\begin{aligned}\frac{df_{xm-1}(u_m)}{du} &= p, \quad \frac{df_{ym-1}(u_m)}{du} = q \\ \frac{d^2 f_{xm-1}(u_m)}{du^2} &= k, \quad \frac{d^2 f_{ym-1}(u_m)}{du^2} = l\end{aligned}$$

Dieser Algorithmus ist in *Matlab*® unter der Funktion *spline* implementiert und wird in einem Programm zur Berechnung der Balltrajektorie verwendet. Der Quellcode ist in Kapitel 5 enthalten.

Abbildung 3.1 zeigt im Vergleich die Balltrajektorie als Gerade bestimmt über lineare Regression (blau) und als Kurve bestimmt über C-Splines (rot). Deutlich zu sehen ist die stetig zunehmende Abweichung, je weiter die Balltrajektorie fortschreitet.

Alternativ kann für eine schnellere Berechnung die Balltrajektorie als Parabel, berechnet aus nur drei Ballposition, implementiert werden (grün). Die Abweichung zum C-Spline ist minimal.

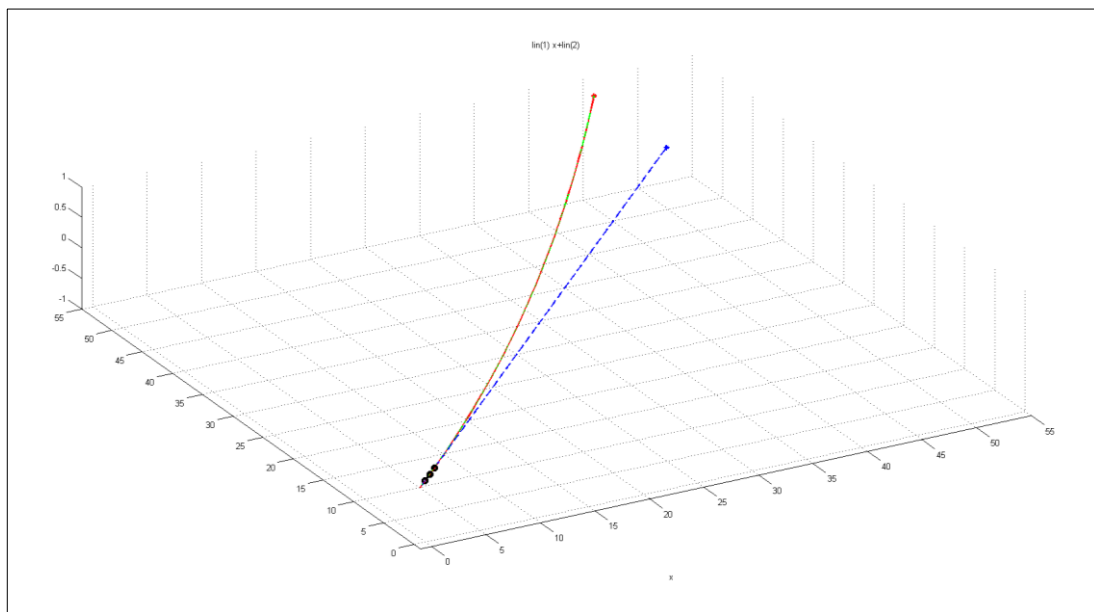


Abbildung 3.1: Vergleich der Algorithmen

### 3.3 Vektorfeld

Der verbesserte Algorithmus zur Berechnung der Balltrajektorie, basierend auf C-Splines, ist jedoch noch immer nicht realitätsgetreu. Dazu müssen weitere physikalische Gegebenheiten in den Algorithmus einfließen.

Die größte Beeinflussung der Balltrajektorie stellt neben dem nun berücksichtigten Drall des Balls die heterogene Spielfeldoberfläche des Tischkickers dar. Um diese Unebenheiten mit einzubeziehen muss ein topologisches Profil des Spielfelds erstellt werden, beispielsweise mittels Laserscan.

Aus diesem Profil wird ein Vektorfeld erstellt, das die Unebenheiten repräsentiert. Dem Vektorfeld kann nun die mit C-Splines berechnete Balltrajektorie superpositioniert werden

und in einem iterativen Verfahren der weitere Verlauf und schließlich die Torwartachssollposition berechnet werden.

Beispielhaft ist in Abbildung 3.2 ein Vektorfeld (grau), die mit C-Splines berechnete Balltrajektorie (blau) und die iterativ mit dem Vektorfeld verrechnete Balltrajektorie (rot) dargestellt.

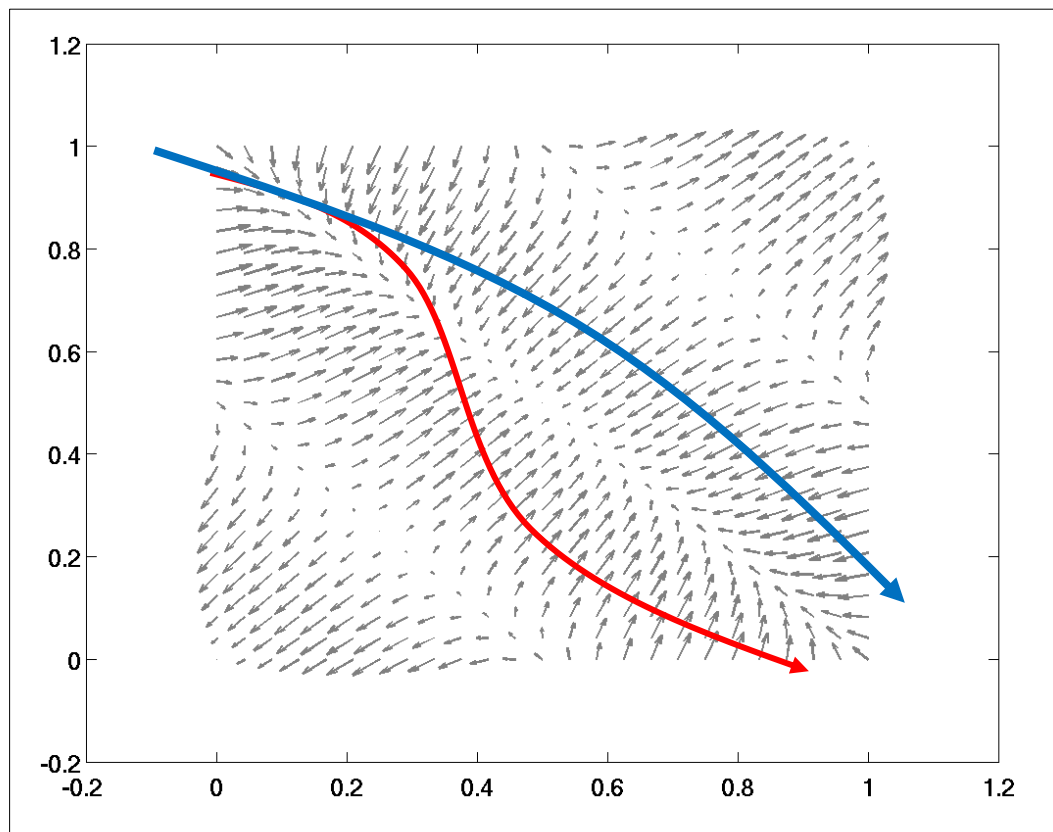


Abbildung 3.2: Balltrajektorie in Vektorfeld

## 4 Ausblick

- Der in Kapitel 3.2 erarbeitete Algorithmus kann in seiner vereinfachten Form als Parabelberechnung aus drei Punkten für erste Tests implementiert werden.
- ... oder direkt als Berechnung mittels C-Splines.
- Vermessung der Spielfeldoberfläche und erstellen des Vektorfelds.
- C-Spline Balltrajektorie mit Vektorfeld superpositionieren und Algorithmus implementieren.

## 5 Anhang

Dieses Kapitel beinhaltet die Anhänge.

## 5.1 Quellcode Matlab®

```
% -----
% Image Understanding SS15
% -----
% Automated Soccer Table
% -----
% Enhanced trajectory prediction
% using c-spline calculation.
% -----

warning off;
clear;
clc;
close all;
figure;
hold on;

% ball positions
x = [0 1 2 ];
y = [0 1 2.01 ];

% target x = goal keeper
xt = 50;

% c-spline
cs = spline(x, y)

% plot positions
plot(x, y, '--ok', 'LineWidth', 3);

% plot c-spline
xx = linspace(-1, xt, 50);
plot(xx, ppval(cs, xx), 'r', 'LineWidth', 2);

% plot target x on spline
plot(xt, ppval(cs, xt), '*r', 'LineWidth', 2);

% fit points on parabel
poly = polyfit(x, y, 2)

% function from poly coeffs
par = @(x) poly(1)*x^2+poly(2)*x+poly(3);

% plot par
h = ezplot(par, [-1,xt]);
set(h, 'color', 'g', 'LineWidth', 2, 'LineStyle', '--');

% delta spline to par
dparcs = par(xt)-ppval(cs, xt)

% target x with linear regression
lin = polyfit(x, y, 1)

% line from coeffs
linreg = @(x) lin(1)*x+lin(2);

% plot line
h = ezplot(linreg, [-1, xt]);
set(h, 'color', 'b', 'LineWidth', 2, 'LineStyle', '--');

% plot target x on line
plot(xt, linreg(xt), '*b', 'LineWidth', 2);

% delta line to spline
dlinregcs = linreg(xt)-ppval(cs, xt)
```

## 6 Literaturverzeichnis

<sup>1</sup> Thesis Mechatronik Bachelor „Konzeption und Realisierung eines automatischen Torwarts für ein Tischfußballspiel – Teil 1: Bildverarbeitung“, Mourad Meziane, 23.01.2009

<sup>2</sup> Thesis Mechatronik Bachelor „Konzeption und Realisierung eines automatischen Torwarts für ein Tischfußballspiel – Teil 2: Motion Control“, Gerald Holz, 22.01.2009