Der Xylophon-Spieler

Duc Manh Pham, Elektromobilität
Otto-von-Guericke-Universität Magdeburg

***Zusammenfassung*—In diesem Projekt geht es um den Bau eines Roboters, dessen Aufgabe es war, Melodien zu spielen. Am Ende des Projektseminars ist es gelungen, ein Roboter mithilfe Lego-Bausteine zu bauen. Der in MATLAB programmierte Roboter war in der Lage, einfache Melodien auf einem Xylophon fließend zu spielen.**

***Schlagwörter*—Lego Mindstorms, MATLAB, Melodien, Roboter, Xylophon.**

I. Einleitung

MENSCHEN haben Musik und Musikinstrumente seit tausenden Jahren gespielt. In Prinzip kann jeder Gegenstand, der Töne oder auch nur Geräusche hervorbringt, als Musikinstrument dienen. [1]

Das Xylophon gehört zu den klassischen Musikinstrumente. In vielen außereuropäischen Musikkulturen wie Asien und Afrika nehmen Xylophone eine wichtige Stellung ein. Vom klassischen Xylophon wurden zahlreiche Arten von Xylophonen weltweit verbreitet. Zum Beispiel: das Balophon in Westafrika, die Marimba in Guatemala, das Trogxylophon pattala in Myanmar, usw. [2]

Roboter und Musikinstrumente haben einen Zusammenhang. Die sind alle menschliche Erfindungen. Man kann einen Roboter konstruieren, um einem bestimmten Zweck zu dienen. Wenn der Zweck ist, Musik zu machen, sollte das nicht unmöglich sein. Trotzdem tauchten viele Probleme während des Baus auf. Der erste Prototyp entstand nach einer Woche Arbeit (siehe Abbildung 1). Dabei sind einige Anforderungen entstanden.

1. Änderung der Positionierung: Der Roboter stand auf dem Tisch ohne feste Position. Der Abstand zwischen Roboter und Xylophon war nicht nahezu konstant. Außerdem wurde der Schläger noch nicht mit dem Oberteil des Roboters befestigt. Es war erforderlich, dieses Problem zu lösen, sonst war der Roboter nicht funktionsfähig.

2. Zuordnung der Tonhöhe zur Winkelposition des Motors: Die Abweichung war relativ groß. Die Töne konnten nicht richtig gespielt werden. Die Ansteuerung war unstabil und ungenau. Der Grund dafür war der NXT-Motor. Der hat bei jedem Befehl etwas falsch durchgeführt. Zum Beispiel: Der Motor sollte sich 90 Grad umdrehen, drehte sich aber einmal um 88 Grad, ein anderes Mal um 92 Grad. Es führte dazu, dass die falschen Klangstäbe geschlagen wurden.

3. Stabilität: Der ganze Roboter hat stark gewackelt, wenn er schnell gedreht wurde. Die zwei Motoren vibrierten viel bei der Ausführung. Man muss den Roboter umbauen, um das Wackeln zu minimieren.

4. Der Roboter sollte nicht nur einzelne Noten abspielen, sondern auch beispielsweise eine ganze Melodie darstellen. Das war eine Anforderung des Programmierens, weil der Roboter einige Prinzipien wie Tonlängen und Tonhöhen verstehen musste.

DOI: 10.24352/UB.OVGU-2020-041

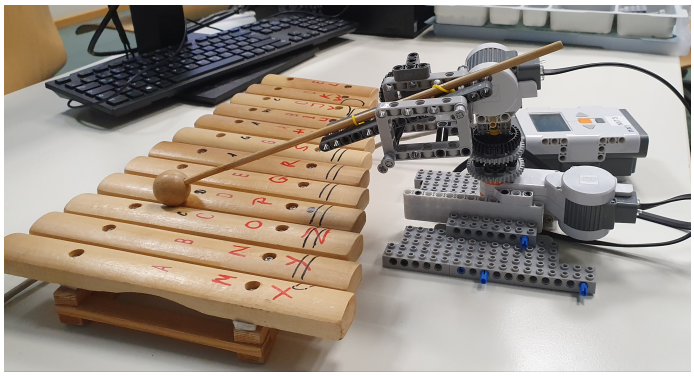

Abbildung 1. Erster Prototyp

II. Vorbetrachtungen

A. Positionierung

Die Probleme mit der Positionierung, die beim ersten Prototyp entstanden sind, wurden in der letzten Version gelöst. Das ganze System von Xylophon und dem Roboterarm wurde mit Lego-Bausteinen auf einer großen flachen Platte festgemacht. Die angemessene Position des Xylophons wurde markiert. Dadurch wurde der Abstand nahezu konstant gehalten. Der Oberteil des Roboterarms wurde neu gebaut, damit sich der Schläger zusammen mit dem Arm bewegte. Das heißt, der Schläger dreht mit gleichem Winkel wie der Roboterarm. Das verhindert, dass sich der Schläger manchmal nicht mit bewegt oder wackelt.

B. Zuordnung der Tonhöhen

Bei der ersten Version konnte der Roboter schon einzelne Note spielen. Aber der Versuch eine Oktave vorzuführen ist noch nicht gelungen, weil die Abweichung nach jeder Drehung zunimmt. Wenn der Arm von C zu C drehen und bei jeder Note einmal schlagen sollte, drehte er von C bis E in der nächsten Tonleiter. Außerdem konnte der Roboter nicht wissen, wo er gerade war. Bei jeder Ausführung musste er zurück zu C per Hand getrieben werden.

Es gab in der letzten Version einige Verbesserungen. Die Genauigkeit stieg deutlich im Vergleich mit dem ersten Prototyp. Der Arm konnte schon zu irgendeinem Klangstab kommen, egal wo er vorher war. Zu Beginn jeder Ausführung ist eine Initialisierung erforderlich, aber sie erfolgt mit drei einfachen Tasten.

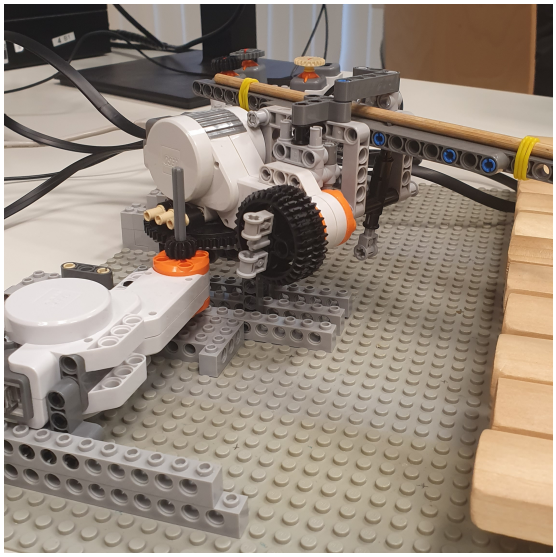

Abbildung 2. Umbau mit einem Drehkranz

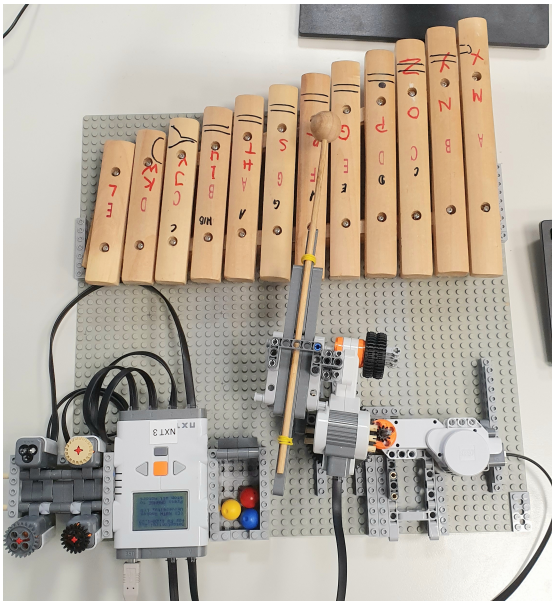

Abbildung 3. Letzte Version

C. Stabilität

Dank eines Videos über ein ähnliches Projekt [3]und der Hilfe des Betreuers konnte der Aufbau des Roboters verbessert werden. Dadurch stieg auch die Stabilität. Ein Drehkranz wurde zwischen zwei Motoren eingefügt, um das Übersetzungsverhältnis zu vergrößern. Das heißt, der in horizontale Richtung bewegende Motor muss um einen größeren Winkel drehen, um das Ziel zu erreichen. Die Abweichung nach mehreren Drehungen wurden dadurch verkleinert (siehe Abbildung 2).

D. Programmieren

Das Programm zur Bedienung des Roboters wurde in Matlab geschrieben. Es wurde dann im Gerät Mindstorms NXT 2.0 übertragen und ausgeführt. Eine zusätzliche Benutzeroberfläche wurde auch programmiert, damit der Roboter einfacher bedient wurde. Ganze Melodien darzustellen war für ihn machbar, da ein Farbsensor angebaut wurde. Je nach Erkennung eine Farbe wurde die entsprechende Melodie abgespielt. Es gab drei verschiedene Melodien zur Auswahl.

Es ist gelungen, die Prinzipien der Musik im Programm mitzubringen. Nicht nur Noten, sondern auch Tonhöhe und ihren Winkelpositionen wurden berücksichtigt und definiert. Eine Korrektur wurde auch hinzugefügt, um einige Fehler bei der Ausführung zu verringern.

III. Realisierung

A. Technischer Aufbau

Wie in Abbildung 3 zu sehen ist, handelt es sich um drei Bereiche. Im oberen Bereich befindet sich ein Xylophon. Das ist ein aus Holz ganz normales Xylophon zum Spielen. Es gibt zwölf Tonhöhen zur Verfügung, aber der Roboterarm konnte nur von C bis C in der nächsten Tonleiter eintreffen. Der Grund dafür ist, damit der Schläger in eine Umlaufbahn fährt, die ähnlich wie ein Kreis ist. Außerdem besteht es einen Vorteil, dass der Winkel zwischen zwei nebeneinander Tonhöhe immer einen bestimmten Wert hat. Die Position des Xylophons wurde mit einigen Lego-Bausteine markiert, damit der Wert nicht verändert wurde.

Im unteren linken Bereich sind Plätze für das NXT-Gerät, einen kleinen Kasten mit drei Kugeln und ein Bedienfeld. Das NXT-Gerät wurde mit zwei NXT-Motoren und vier Sensoren verbunden. Alle Programme zur Vorführung des Roboters laufen in einem Rechner. Die Signale oder Befehle wurden dann über ein USB-Kabel zum NXT-Gerät gesendet. Danach wurde alles ausgeführt. Der kleine Kasten war da, um die farbige Kugeln zu lagern und um den linken Bereich auf der Platte zu befestigen. Es gibt im Bedienfeld vier Sensoren: drei Tastsensoren und ein Farbsensor. Der Farbsensor dient dazu, dass man verschiedene Melodien für den Roboter auswählen kann, welche er dann abspielt. Drei Tastsensoren sind für den Handantrieb da. Wird die graue (schwarze) Taste gedrückt, bewegt sich der Arm nach links (rechts). Wird die gelbe Taste gedrückt, dreht der in vertikale Richtung bewegende Motor. Dieses Bedienfeld dient zur Initialisierung. Zu Beginn jeder Ausführung muss der Roboterarm zur Klangplatte C ausgerichtet werden. Der Kugel auf dem Schläger muss genau mittig auf C angeordnet werden. Der Stab, der mit dem zweiten Motor verknüpft wurde, muss senkrecht nach oben zeigen. Je genauer diese Schritte durchgeführt werden, desto besser wird das Ergebnis. Um die Initialisierung zu beenden, werden die graue Taste und die schwarze Taste gleichzeitig gedrückt.

Im rechten Bereich platziert sich der Roboter. Er wurde nach mehreren Prototypen verbessert. Solche Probleme wie Wackeln und Erschütterung kommen selten vor, da er ziemlich fest auf der Platte steht. Der Roboter basiert auf zwei Motoren. Der einzige Zweck ist, dass der Schläger so frei wie möglich in horizontale und vertikale Richtung gedreht werden kann. Beim ersten Prototyp wurden zwei Motoren direkt miteinander verbunden. Wird der Wert von „Tacho Limit“ gleich 10 im Matlab-Programm gespeichert, wird der Schläger genau 10

Grad gedreht. Da der Abstand zwischen zwei Tonhöhen relativ klein ist, passierte es oft so, dass falsche Klangstäbe geschlagen wurden und die Melodie verdorben wurde. Dieses Problem wurde großenteils gelöst, indem man ein System mit zwei Zahnräder gebaut hat. Das größere Zahnrad, nämlich der Drehkranz, wurde sich an dem zweiten Motor angelagert. Das kleinere Zahnrad fügte man an den ersten Motor über einen Stab an. Das hat eine Wirkung, dass der Schläger deutlich kleinerer Winkel drehen lässt, als den Drehwinkel des ersten Motors. Dadurch wird die Chance, dass falsche Klangstäbe geschlagen werden, geringer. Zusammenfassend lässt sich sagen, dass die Spiel-Genauigkeit besser wird.

B. Programmablauf

In der Abbildung 4 wird gezeigt, wie das Hauptprogrammm geschrieben wurde. Zuerst muss der Schalter umgelegt werden. Dann ist eine Initialisierung erforderlich. Das macht man mit Tasten am Bedienfeld. Wenn die Initialisierung erfolgt, kommt danach die Songauswahl. Wird eine Farbkugel am Farbsensor gebracht, beendet die Auswahl. Eine Nachricht wird auf dem Bildschirm angezeigt und die dazugehörige Melodie wird abgespielt.

Die Abbildung 5 legt den Ablaufplan des Programm dar, welches den Farbsensor regelt. Hier wurden Switch-Verzweigungen verwendet, die dann entschieden, welche Melodie zur Vorführung gewählt wurde. Aus zeitlichen Gründen konnten nur drei Melodien programmiert werden. Rote Kugel war für den Song „Summ summ summ“, blau für „Alle meine Entchen“ und gelb für „Twinkle twinkle little star“.

C. Logische Steuerung

Es ist gelungen, die Ideen bei der Kick-Off-Präsentation im Programm mitzunehmen. Die Töne und Tonlängen wurden in zwei Felder gespeichert (siehe Anhang). Der Abstand zwischen zwei nebeneinander Klangstäbe ist 11 Grad. Mit Faktor drei des Übersetzungsverhältnisses ist der Abstand der Töne verdreifacht (33 Grad). Jede einfache Melodie enthält Viertelnote, halbe Note und ganze Note. Die Dauer der Viertelnote ist 45 ms, halbe Note ist zweimal Viertelnote und ganze Note ist gleich zweimal halbe Note. Durch Winkelpositionen des Schlägers wurden Töne definiert. Eine Korrektur wurde auch hinzugefügt. Wenn der Schläger einen langen Weg fahren muss, wie beispielsweise die Länge von vier Klangstäben, dann muss er noch länger als üblich fahren, damit er das Ziel erreicht. Die Korrektur ist von Melodie zu Melodie unterschiedlich. Das Programm wurde in einer for-Schleife gemacht, weil es viele Vorteile bot. Für unterschiedlichen Melodien wird die selbe for-Schleife verwendet. Nur das Ton-Feld und das Tönlänge-Feld muss geändert werden. Das spart Zeit beim Programmieren, weil kein ganz neues Programm geschrieben werden muss. Die Länge der Melodie spielt keine große Rolle mehr. Das Programm wird nicht länger und übersichtlich. MATLAB ist ein Interpreter. Je kürzer die Übersetzung dauert, desto schneller wird die Ausführung. Deswegen konnte der Roboter noch früher in Bewegung setzen.

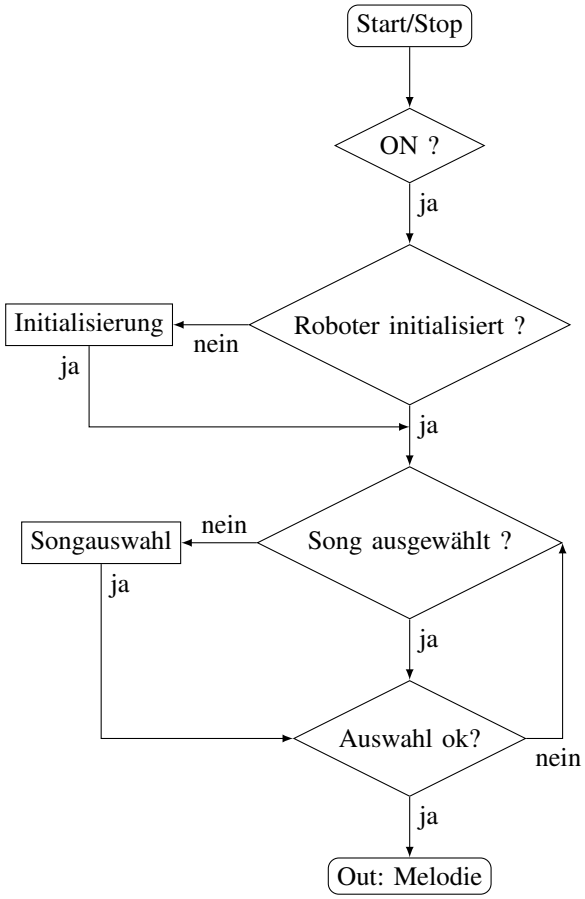

Abbildung 4. Programmablaufplan für die Benutzeroberfläche

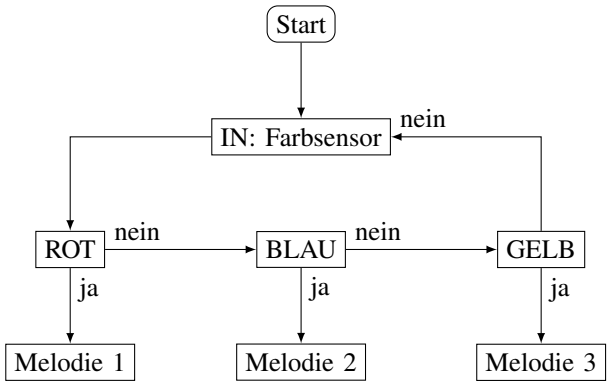

Abbildung 5. Programmablaufplan für Auswahl der Melodien durch Farbsensor

IV. Ergebnisdiskussion

Wie bei der Abschlusspräsentation oder in Youtube [4] zu gucken ist, funktionierte der Roboter tatsächlich schon. Es ist auch erkennbar, dass welche Melodie er gerade darstellt. Trotzdem gibt es auch Probleme, die man nicht vermeiden kann. Die Durchschnittsgeschwindigkeit war noch langsam, im Vergleich mit einem normalen Spieler. Es klingelte nicht schön wie es sein soll. Die Melodien wurden extra gewählt, damit sich der Roboter am wenigsten bemühen muss. Aber das Ergebnis war nicht so toll wie erwartet. Komplexe Melodien sind folglich für den Roboter nicht spielbar. Noch ein Problem

tauchte bei der Vorführung auf, als er gerade eine Melodie spielte, gab das NXT-Gerät ein Ton von sich. Das passierte, weil zu viele Befehle auf einmal gekommen sind. Das Programm muss im Prinzip noch optimiert werden. Die Benutzeroberfläche funktionierte nicht einwandfrei. Bei jeder Ausführung konnte nur eine Melodie ausgewählt werden. Als man die Wahl ändern wollte, funktionierte das Programm nicht mehr.

V. Zusammenfassung und Fazit

Als Fazit lässt sich sagen, dass das Projekt schon in die Richtung fuhr. Mit Maschinen Musik zu machen ist auf jeden Fall eine interessante Idee. Nach der zweiwöchigen Arbeit wurde ein Produkt bereitgestellt und präsentiert. Aber das Endprodukt hätte nicht entstehen können, wenn es keine Mühen von Beteiligten und Hilfen von Betreuers gäbe. Es ist allerdings erforderlich, Grundkenntnisse über MATLAB zu haben und kreativ zu sein. Der Roboter hat noch Potential, bestehende Probleme zu lösen und besser, menschlicher zu spielen. Ein weiterer Motor könnte hinzugefügt werden, um die Leistung und die Geschwindigkeit zu steigern. Das Programm könnte verkürzt und logischer strukturiert werden. Die Anwendungsgebiet könnte in einem Konzert oder in einer Musikschule liegen, aber der Weg bis dahin ist noch so weit.

Anhang

Im Anhang wird der Quelltext des Programms, der den Roboter regelt, um Musik zu machen, gezeigt. Die Aufklärung des Programms befindet sich im Unterabschnitt III.C Logische Steuerung.

```
l = 33; %Abstand von einer Tonhöhe
v = 445/1000; %Viertelnote
h = 2 * v; % Halbenote
g = 2 * h; %ganze Note
tonlaenge = [v,v,v,v,h,h,v,v,v,v,g,v,v,v,v,g,v,v,v,v,h,h,v,v,v,v,g];
toene = [0,1,1,1,1,0,1,0,0,0,-1,1,0,0,0,-1,-1,0,0,0,-1,0,-1,0,0,0,-1];
%Melodie-Schleife
for i = 1:length(toene)

        %Anfahren der Töne
        if (toene(i) == 0) %gleicher Ton
         %nichts passiert
        elseif(toene(i) < 0) %tieferer Ton
             motor_h.Power=100 * -1;
             motor_h.TachoLimit= abs(toene(i)) * l - 1;
             motor_h.SendToNXT();
        else
             motor_h.Power=100; %höherer Ton
             motor_h.TachoLimit= abs(toene(i)) * l;
             motor_h.SendToNXT();
        end
        % Spielen der Töne
        pause(745/1000);
        motor_v.Power=100;
        motor_v.TachoLimit=180;
        motor_v.SendToNXT();
        pause(tonlaenge(i));
end
end
```

Literaturverzeichnis

[1] Wikipedia: *Musikinstrument*, Dezember 2019. http://https://https://de.wikipedia.org/wiki/Musikinstrument

[2] Wikipedia: *Xylophon*, Februar 2020. http://https://de.wikipedia.org/wiki/Xylophon

[3] Kanal Technew Robot, Youtube: *Lego EV3 robot plays music on a glockenspiel xylophone*, https://www.youtube.com/watch?v=-FYDRMKozoc

[4] Kanal Mathias Magdowski, Youtube: *Xylophon-Roboter aus dem Lego-Praktikum 2020 an der OVGU Magdeburg spielt "Twinkle Little Star"*, https://www.youtube.com/watch?v=zFOnkqJ9IaQ