

**AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE**



Wydział Elektrotechniki, Automatyki, Informatyki
i Inżynierii Biomedycznej
Katedra Automatyki i Inżynierii Biomedycznej

PRACA DYPLOMOWA
Magisterska

**System zdalnego sterowania modelem helikoptera z
wykorzystaniem wizyjnego sprzężenia zwrotnego**

Helicopter's remote control system using video feedback

Imię i nazwisko: Rafał Drużyński
Kierunek studiów: Automatyka i Robotyka

Opiekun pracy Dr inż. Paweł Rotter

Kraków 2013

Oświadczam, świadomy odpowiedzialności karnej za poświadczenie nieprawdy, że niniejszą pracę dyplomową wykonałem osobiście i samodzielnie i że nie korzystałem ze źródeł innych niż wymienione w pracy.

Serdecznie dziękuję
Panu dr inż. Pawłowi Rotterowi
za wyrozumiałość i pomoc w napisaniu niniejszej pracy

SPIS TREŚCI

1.	Wstęp	5
1.1	Cel pracy	5
1.2	Układ pracy	6
2.	Opis zagadnienia	7
2.1	System	7
2.2	Opis helikoptera	9
3.	Układ obserwacji	11
3.1	Wstęp	11
3.2	Podstawy teoretyczne	12
3.3	Układ stereowizyjny	16
3.4	Obserwacja położenia	20
3.5	Obserwacja orientacji	23
4.	Testy	29
4.1	Test algorytmu wykrywania diody	29
4.2	Test algorytmu wykrywania diody z odejmowaniem tła	34
4.3	Test działania stereowizji	36
4.4	Test algorytmu wykrywania prostokąta	39
5.	Moduł komunikacji	49
5.1	Wstęp	49
5.2	Odbiornik IR	50
5.3	Nadajnik IR	55
5.4	Analiza protokołu komunikacji	58
5.5	Plik konfiguracyjny	65
5.6	Próba nawiązania komunikacji	67
6.	Model symulacyjny	72
6.1	Model matematyczny helikoptera	72
6.2	Algorytm rozwiązywania równań różniczkowych	75
7.	Algorytmy sterowania	76
7.1	Wstęp	76
7.2	Sterowanie neutralne	77
7.3	Algorytm stabilizacji w punkcie	79
7.4	Algorytm lotu do punktu	83
8.	Algorytm sterowania optymalnego	87
8.1	Sformułowanie problemu optymalizacji i jego analiza	87

8.2	Algorytm optymalizacji gradientowej.....	94
8.3	Testy algorytmu optymalizacyjnego	96
8.4	Porównanie algorytmów	102
9.	Podsumowanie	105
10.	Bibliografia.....	108
11.	Załączniki	108
	Dodatek A – Spis materiałów zawartych na płycie	109

1. Wstęp

1.1 Cel pracy

Tematem pracy jest stworzenie systemu pozwalającego na zdalne sterowanie modelem helikoptera. W skład projektu wchodzi konstrukcja układu umożliwiającego odczyt położenia i orientacji helikoptera oraz elementu nadawczego, którego zadaniem jest komunikacja z helikopterem. Elementem pracy jest także implementacja algorytmów pozwalających na sterowanie systemem. Na potrzeby opracowania stworzono model matematyczny pozwalający na teoretyczną analizę zagadnienia oraz zaimplementowano algorytm BFGS wyszukujący sterowanie optymalne dla helikoptera przy zadanych warunkach początkowych z wykorzystaniem zasady maksimum Pontriagina.

1.2 Układ pracy

Pierwsza część zawiera krótkie wprowadzenie zawierające cele pracy oraz przedstawienie układu poszczególnych rozdziałów.

W drugim rozdziale pracy bardziej szczegółowo przedstawiono charakterystykę omawianego zagadnienia. W tej części pracy pokazano podejście do problemu, jakie było w dalszej jej części egzekwowane oraz podano przyczyny takiego wyboru.

Trzecia część przedstawia stworzony system obserwacji. Część ta zawiera niezbędne informacje teoretyczne, a także wszystkie algorytmy biorące udział w obróbce obrazu. Wyniki uzyskane w wyniku obróbki obrazu zostały omówione w kolejnym rozdziale.

Rozdział piąty przedstawia stworzony system nadawczy. Rozdział ten zawiera niezbędne informacje teoretyczne oraz schematy niezbędne przy konstrukcji modułu, a także pokazuje, w jaki sposób zidentyfikowano protokół komunikacyjny. Ostatnia część rozdziału zawiera opis podjętych prób nawiązania komunikacji z helikopterem.

Następny rozdział zawiera opis stworzonego modelu symulacyjnego wraz z algorytmem symulującym jego zachowanie.

Kolejna część przedstawia proste algorytmy zaimplementowane na potrzeby pracy wraz z ich objaśnieniem i przetestowaniem na modelu symulacyjnym.

Rozdział ósmy zawiera wszelkie informacje teoretyczne oraz implementacje algorytmu sterowania bazującego na zasadzie maksimum Pontriagina. Przedstawia także omówienie uzyskanych dzięki temu wyników oraz porównanie ich z wynikami uzyskanymi dzięki algorytmom z poprzedniego rozdziału.

W ostatnim rozdziale podsumowano całą pracę oraz przedstawiono wnioski wynikające z analizy omawianego zagadnienia wraz z przedstawieniem ścieżek ewentualnego rozwoju.

2. Opis zagadnienia

2.1 System

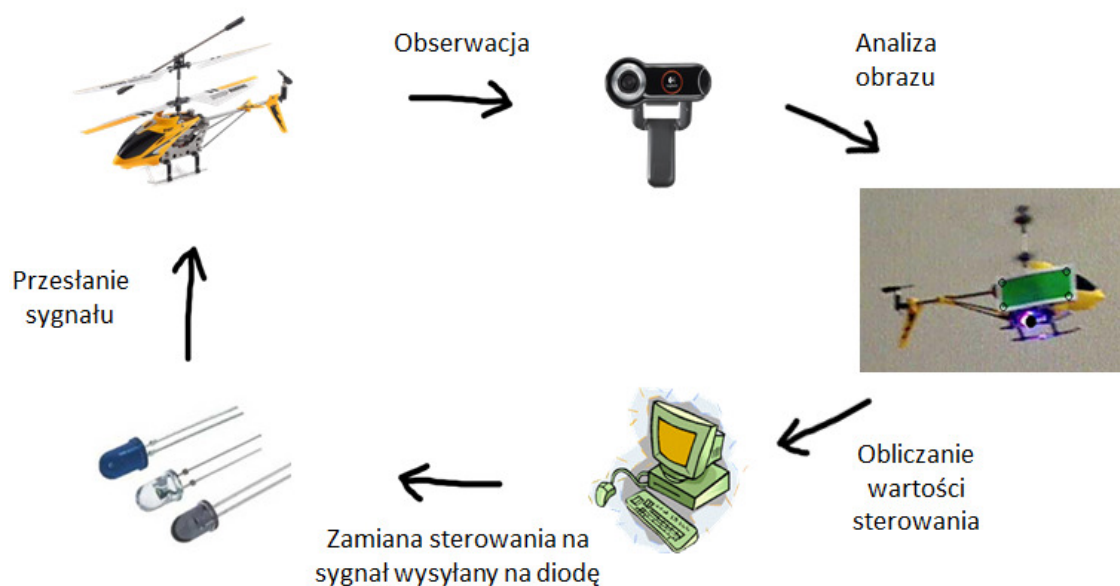
Dzięki rozwojowi techniki i technologii w życiu codziennym napotykamy na rozwiązania do niedawna używane jedynie przez wojsko czy inne agencje rządowe. Przyzwyczajamy się do otaczających nas rzeczy bez większego zastanowienia nad zasadą ich działania, czy problemami, z jakimi musieli spotkać się konstruktorzy. Często bywa tak, że to, co teraz jest zabawką jeszcze 10 lat temu było częścią ściśle tajnego projektu. Nie zauważając tego szukamy problemów, czy interesujących zagadnień bardzo daleko, a nieczęsto ciekawe mamy pod ręką

Inspiracją tej pracy jest właśnie problem na bazie zabawki – zdalnie sterowanego helikoptera. Istotą jest stworzenie systemu pozwalającego na sterowanie helikopterem nie za pomocą pilota – jak to w oryginale zapewnia sprzedawca – ale za pomocą komputera. Układ taki pozwalałby na ruch śmigłowca bez ingerencji człowieka.

Pomysł taki napotyka jednak na kilka problemów. Z helikoptera nie można po prostu poprowadzić kabla do komputera tak by ten wiedział o nim wszystko. Dodatkowo niewielkie wymiary uniemożliwiają zastosowanie jakichkolwiek czujników na jednostce. Jedynym rozwiązaniem nieingerującym w śmigłowiec jest system wizyjny. Ponieważ ruch odbywa się w 3 płaszczyznach zastosowano stereowizyjny układ kamer podłączonych do komputera.

Kolejny problem występuje przy przesyłaniu sygnału do helikoptera. Ponownie problemem jest ewentualne poprowadzenie kabla, a bezprzewodowy odbiornik wi-fi czy bluetooth waży więcej niż zastosowany model. Jedynym rozwiązaniem jest wykorzystanie elektroniki oryginalnie zamontowanej na śmigłowcu – odbiornika podczerwieni. Takie rozwiązanie wymagało zbudowania w ramach pracy nadajnika z nim współpracującego.

Po uwzględnieniu wszystkich powyższych konkluzji system konstruowany w pracy można przedstawić w postaci schematu zaprezentowanego poniżej.



Rysunek 2.1- Schemat ideowy systemu
Opracowanie własne na podstawie syma.com, logitech.com

Schemat ten przedstawia jedynie ideę, zgodnie z którą powstawał cały układ. Kolejne elementy systemu zostały opisane w dalszej części pracy.

2.2 Opis helikoptera

Głównym elementem systemu konstruowanego na potrzeby pracy jest zdalnie sterowany helikopter. W pracy użyto śmigłowca SYMA s107 widocznego poniżej.



Rysunek 2.2 - Helikopter SYMA s107

Źródło: symba.com

Helikopter ten charakteryzuje się niewielkimi rozmiarami – długość od przedniej części kadłuba do tylnego śmigła wynosi zaledwie 16 cm. Szerokość płóz wynosi zaledwie 4 cm. Tak małe rozmiary skutkują oczywiście niską wagą jednostki – jedynie 34 g. Przy takiej wadze wzrasta podatność na czynniki zewnętrzne – przeciąg lub nawet słabe dmuchnięcie powoduje wytrącenie z równowagi

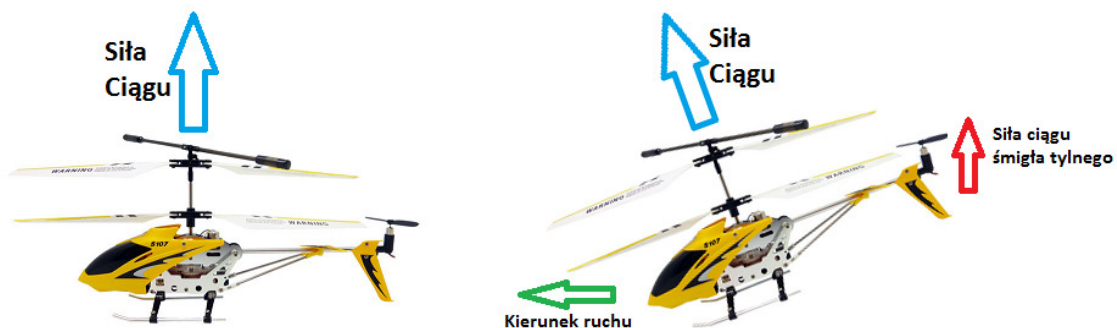
Śmigłowiec sterowany jest przy użyciu sygnału podczerwonego – producent umieścił w modelu odbiorczą diodę IR. Sterowanie odbywa się za pomocą pilota, dzięki któremu można przesyłać 3 informacje – lot w górę, obrót wokół osi pionowej i lot do przodu/do tyłu.

Dodatkowo helikopter jest wyposażony w zwykłą diodę LED ustawioną w części przedniej kadłuba i skierowaną w dół. Dioda ta miga naprzemiennie na czerwono i niebiesko.

Jedną z ważniejszych rzeczy, które charakteryzują użyty model jest bardziej sztywna konstrukcja w porównaniu do konwencjonalnego helikoptera. Zamiast 6 stopni swobody występują tylko 4.

W większości śmigłowców występuje tzw. układ Sikorskiego – charakteryzuje się on występowaniem tylko jednego śmigła górnego. Posiada ono natomiast zmienny kąt natarcia łopatek, co w połączeniu ze sterowaniem zależnym od położenia pojedynczej łopaty daje możliwość skierowania siły w dowolnym kierunku. Tylne śmigło ustawione jest w osi prostopadłej do górnego i odpowiada za stabilizację i obrót wokół osi pionowej.

Inna sytuacja występuje w helikopterze firmy SYMA. Został on skonstruowany w oparciu o układ Kamowa. Posiada on dwa śmigła górne o stałym kącie natarcia i przeciwnych kierunkach obrotu. Odpowiadają one za lot w górę, a różnica w ich prędkościach powoduje obrót wokół osi pionowej. Tylne śmigło ustawione jest w takiej samej płaszczyźnie i ma za zadanie podnosić bądź opuszczać tył śmigłowca. Takie wychylenie powoduje zmianę kierunku siły ciągu górnych śmigieł, co skutkuje ruchem postępowym. Można to zaobserwować na poniższym schemacie.



Rysunek 2.3- Zasada powstawania ruchu postępowego

Opracowanie własne na podstawie syma.com

W wielu nowoczesnych jednostkach wykorzystywany jest właśnie taki układ, ale z pominięciem śmigła tylnego. Dzięki takiemu rozwiązaniu zmniejszone zostają rozmiary helikoptera, a może osiągać on bardzo wysokie prędkości – najszybszy aktualnie śmigłowiec zbudowany został właśnie na tej zasadzie. Rozwiązanie takie ma jednak swoje wady – wprowadza drgania, zmniejsza sterowność w zawisie i wymaga bardzo skomplikowanego układu sterowania.

3. Układ obserwacji

3.1 Wstęp

Obiekt regulacji, jakim jest helikopter ma jedną bardzo istotną wadę – w żaden sposób nie dostarcza nam żadnych informacji zwrotnych. Z punktu widzenia automatyki dysponujemy obiektem bez wyjść. W takim przypadku stworzenie zamkniętego układu regulacji jest niemożliwe. Śmigłowiec posiada jednak w każdej chwili czasu położenie określone przez 3 zmienne oraz orientację – należało tylko stworzyć układ umożliwiający pozyskanie i przesłanie do komputera tych informacji

Jak już wspomniano małe gabaryty jednostki uniemożliwiają większość rozwiązań. Czujniki, czy jakiegokolwiek dodatkowe urządzenia zbytnio obciążą helikopter, co uniemożliwi mu swobodny lot. Rozwiązaniem, które zastosowano w pracy było stworzenie systemu wizyjnego, który byłby w stanie pozyskać niezbędne informacje do jednoznacznego opisu aktualnego stanu śmigłowca.

W przypadku helikoptera użytego w pracy orientację można opisać używając tylko jednej zmiennej przedstawiającej obrót wokół osi pionowej. Pozostałe dwa kąty nie są istotne – obrót wokół osi przechodzącej wzdłuż śmigłowca nie występuje z racji jego usztywnionej konstrukcji, a trzeci z kątów, czyli pochylanie i podnoszenie dzioba jest niewielki i bezpośrednio realizowany przez sterowanie tylnym śmigłem.

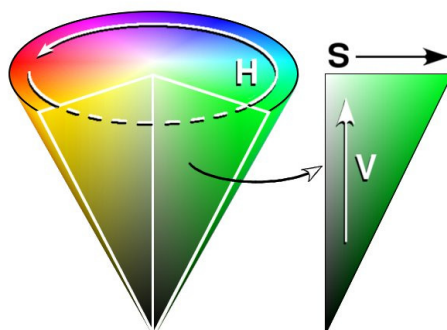
Na potrzeby pracy stworzono układ stereowizyjny składający się z dwóch kamer LOGITECH WebCAM 9000 podłączonych poprzez łącze USB do komputera. Taki układ w połączeniu z odpowiednią obróbką obrazu jest w stanie pozyskać wszystkie niezbędne informacje na temat helikoptera.

Przestawione w kolejnych dwóch podrozdziałach informacje teoretyczne sporządzono w oparciu o [1] - [5].

3.2 Podstawy teoretyczne

Podstawowym zagadnieniem jest rozumienie obrazu z punktu widzenia komputera. Jak wiadomo obraz składa się małych kwadratów zwanych pikselami. Ich ilość określana jest przez rozdzielczość kamery. Im ich więcej tym lepsza jakość, ale także dłuższy czas pozyskiwania obrazu i obróbki. Człowiek widząc na ekranie komputera obraz może zaobserwować kolory poszczególnych pikseli. Można stwierdzić na przykład, że konkretny piksel jest żółty. Komputer jednak nie zapisuje sobie w pamięci, że piksel znajdujący się w konkretnym miejscu jest „żółty”. Jak więc komputer przechowuje takie informacje?

W cyfrowym zapisie obrazu występuje pojęcie przestrzeni barw. W takiej przestrzeni pojedynczy punkt odpowiada konkretnej barwie. Istnieje kilka różnych przestrzeni. Wszystkie jednak posiadają 3 wymiary – oznacza to, że do jednoznacznego opisanie koloru wymagane są 3 wartości. Jedną z popularniejszych przestrzeni barw jest RGB, w której składowymi są kolory: czerwony (R), zielony (G) i niebieski (B) – kolor wynikowy powstaje z pomieszczenia w odpowiednich proporcjach tych właśnie składowych. Ich wartości przedstawiane są najczęściej od 0 do 1 albo, co w komputerach spotyka się częściej od 0 do 255. Kolejną przestrzenią użytą w pracy jest HSV. Pierwszą składową jest odcień (ang. Hue – H), który jest kątem w zakresie od 0 do 359 na kole barw. Kolejnymi składowymi są: nasycenie (ang. Saturation – S) i moc światła (ang. Value – V). Są one kolejno promieniem i wysokością stożka, którego podstawą jest koło barw. Zasada ta zilustrowana jest na rysunku 3.1. Obie wartości mogą być przedstawiane w zakresie od 0 do 1 albo od 0 do 255.



Rysunek 3.1 - Zasada działania przestrzeni barw HSV

Źródło: wikipedia.pl

Podsumowując powyższe informacje każdy piksel jest zdefiniowany przez 3 wartości, co oznacza, że cały obraz przechowywany jest w komputerze jako 3 tablice. W każdej z tych tablic występują najczęściej wartości od 0 do 255. Tablica taka wciąż jest trudna do analizy, dlatego stosujemy przekształcenia zwane obróbką obrazu do otrzymania postaci jak najprostszej. Jedną z bardziej przydatnych transformacji jest binaryzacja. Jak sama nazwa wskazuje obraz po takim przekształceniu posiada tylko dwie wartości. Binaryzacja przypisuje wszystkim elementom o wartościach wyższych niż zadany próg wartość 1, a pozostałym 0. W pracy wykorzystano konwencje przypisującą pikselom z wartością 1 miano obiektu, a pozostałym otoczenia.

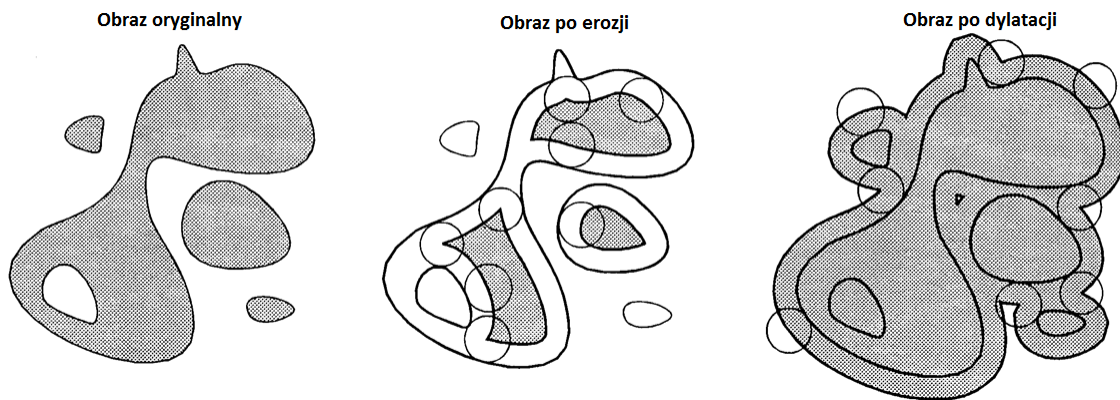


Rysunek 3.2 - Przykład zastosowania binaryzacji

Rysunek 3.2 prezentuje przykład zastosowania binaryzacji z różnymi wartościami progów. Najwyżej widoczny jest obraz oryginalny, a kolejne obrazy są wynikami binaryzacji z rosnącym progiem. Dobór progów może odbywać się na kilka sposobów. Niektóre metody doboru balansują wynikową liczbę pikseli z jedną i drugą wartością. Inne wyszukują lokalne minima histogramu. Prógi takie można też podać wnioskując z doświadczenia.

Ważną rolę dla obrazów binarnych odgrywają przekształcenia morfologiczne. Erozja i dylatacja są właśnie takimi operacjami. Obie metody wykorzystują tzw. element

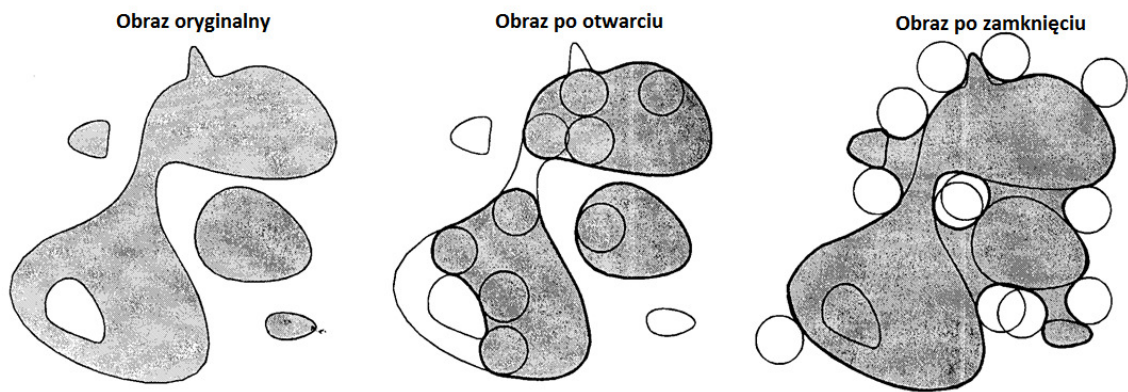
strukturalny. Element ten może być dowolną figurą, jednak najczęściej wykorzystuje się w tym celu koło o zadanym promieniu, bądź kwadrat. Dla obu operacji do każdego punktu obrazu przykłada się środek elementu strukturalnego. W przypadku erozji punkt wynikowy ma wartość 1, jeżeli wszystkie punkty elementu strukturalnego mają wartość 1. Inna sytuacja ma miejsce w przypadku dylatacji – wystarczy, że przynajmniej jeden punkt objęty przez element strukturalny ma wartość 1. Przykłady zastosowania tych metod są widoczne na rysunku 3.3.



Rysunek 3.3 - Przykład zastosowania erozji i dylatacji wraz z zaznaczeniem el. Strukturalnego

Źródło: Przetwarzanie i rozpoznawanie obrazów – wykłady [2]

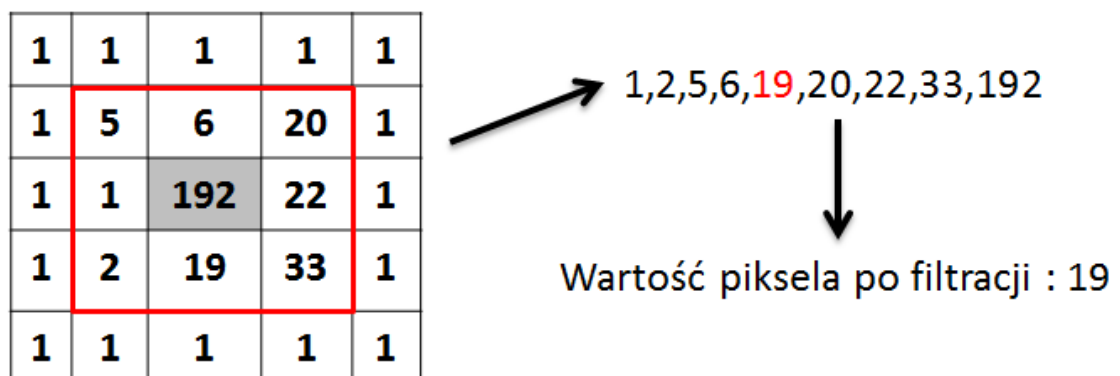
Przydatne w obróbce obrazu są też złożeniach wyżej wymienionych operacji. Dylatacja poprzedzona erozją nosi nazwę otwarcia. Operacja ta nie zmienia wielkości obiektów na obrazie, a także usuwa drobne elementy „wystające” z obiektów. Zamiana kolejności wykonywania operacji daje przekształcenie zwane zamknięciem. Operacja ta podobnie jak otwarcie nie zmienia wielkości obiektów na obrazie równocześnie usuwając dziury w obiektach. Przykłady użycia tych operacji prezentuje rysunek 3.4.



Rysunek 3.4 - Przykład zastosowania otwarcia i zamknięcia wraz z zaznaczeniem el. Strukturalnego

Źródło: Przetwarzanie i rozpoznawanie obrazów – wykłady[2]

Kolejnym ważnym przekształceniem obrazu jest filtracja. Filtracja jest zaliczana do grupy przekształceń kontekstowych – oznacza to, że do uzyskania piksela na obrazie wynikowym potrzebny jest jego odpowiednik z obrazu oryginalnego wraz ze swoim otoczeniem. Do oryginalnego obrazu przykładana jest maska o odpowiednich wymiarach i na jej podstawie obliczana jest wartość piksela wynikowego. Często używane są filtry górnoprzepustowe (wzmacniają małe elementy) oraz dolnoprzepustowe (do eliminacji zakłóceń). Kolejną często używaną metodą jest filtracja medianowa. Pozwala ona na uśrednienie obrazu, przez co można pozbyć się z obrazu szumów przy niewielkiej stracie informacji. Polega ona na pozyskaniu wartości piksela oraz wartości pikseli z jego otoczenia – rozmiar otoczenia zależy od wielkości użytej maski. Następnie z wartości tych wylicza się medianę – wartość ta jest przyporządkowywana pikselowi po przekształceniu.



Rysunek 3.5 - Przykład zastosowania filtru medianowego

3.3 Układ stereowizyjny

W pracy zastosowano stereowizyjny układ kamer. Układ taki składa się z dwóch kamer ustawionych pod różnymi kątami względem środka obserwowanej przestrzeni. Odległości i kąty ustawiania kamer nie mają znaczenia – jedynym kryterium jest dobra widoczność całej przestrzeni lotu helikoptera. Stereowizja opiera się o wielokrotne wykorzystanie transformacji płaskiej.

Transformacja płaska polega na przyporządkowaniu punktom jednej płaszczyzny, punktów leżących na drugiej płaszczyźnie odpowiednio zorientowanej w przestrzeni. Pierwszą z nich zwaną monitorową najczęściej utożsamia się z ekranem bądź matrycą kamery. Druga z nich, ze względu na dowolne usytuowanie w przestrzeni nazywa się lokalną, bądź rzeczywistą. Zależność między współrzędnymi odpowiadających sobie punktów na płaszczyznach można opisać wzorem:

$$TP(P_M) = P_R = \begin{bmatrix} x_R \\ y_R \end{bmatrix} = \begin{bmatrix} \frac{a_1 x_M + b_1 y_M + c_1}{\alpha x_M + \beta y_M + 1} \\ \frac{a_2 x_M + b_2 y_M + c_2}{\alpha x_M + \beta y_M + 1} \end{bmatrix}; \quad P_M = \begin{bmatrix} x_M \\ y_M \end{bmatrix} \quad (3.1)$$

Oznacza to, że do jednoznacznego wyznaczenia takiej transformacji niezbędne jest zidentyfikowanie 8 współczynników : $a_1, b_1, c_1, a_2, b_2, c_2, \alpha, \beta$. Dla uproszczenia oznaczono wektor współczynników przez WSP. Warto zauważyć, że współczynniki te uwzględniają zarówno wzajemne położenie płaszczyzn, jak i parametry optyczne kamery.

Do identyfikacji współczynników niezbędna jest znajomości współrzędnych 4 punktów na płaszczyźnie monitorowej oraz odpowiadających im punktów na płaszczyźnie rzeczywistej. Ponadto żadne trzy ze znanych punktów nie mogą być współliniowe. Podstawiając znane punkty do wzoru (3.1) otrzymano:

$$\begin{bmatrix} x_{R1} \\ y_{R1} \\ x_{R2} \\ y_{R2} \\ x_{R3} \\ y_{R3} \\ x_{R4} \\ y_{R4} \end{bmatrix} = \begin{bmatrix} x_{M1} & y_{M1} & 1 & 0 & 0 & 0 & -x_{M1}x_{R1} & -y_{M1}y_{R1} \\ 0 & 0 & 0 & x_{M1} & y_{M1} & 1 & -x_{M1}x_{R1} & -y_{M1}y_{R1} \\ x_{M2} & y_{M2} & 1 & 0 & 0 & 0 & -x_{M2}x_{R2} & -y_{M2}y_{R2} \\ 0 & 0 & 0 & x_{M2} & y_{M2} & 1 & -x_{M2}x_{R2} & -y_{M2}y_{R2} \\ x_{M3} & y_{M3} & 1 & 0 & 0 & 0 & -x_{M3}x_{R3} & -y_{M3}y_{R3} \\ 0 & 0 & 0 & x_{M3} & y_{M3} & 1 & -x_{M3}x_{R3} & -y_{M3}y_{R3} \\ x_{M4} & y_{M4} & 1 & 0 & 0 & 0 & -x_{M4}x_{R4} & -y_{M4}y_{R4} \\ 0 & 0 & 0 & x_{M4} & y_{M4} & 1 & -x_{M4}x_{R4} & -y_{M4}y_{R4} \end{bmatrix} \cdot \begin{bmatrix} a_{R1} \\ b_{R1} \\ c_{R2} \\ a_{R2} \\ b_{R3} \\ c_{R3} \\ \alpha \\ \beta \end{bmatrix} \quad (3.2)$$

Stosując odpowiednie oznaczenia zapisano w postaci:

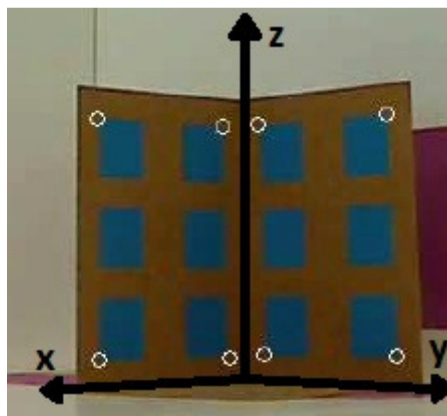
$$H = C \cdot WSP \quad (3.3)$$

Biorąc pod uwagę wcześniej wspomniany warunek o braku współliniowości układ ten posiada tylko jedno rozwiązanie postaci:

$$WSP = H \cdot C^{-1} \quad (3.4)$$

Do uzyskania transformacji płaskiej najłatwiej użyć wzorca kalibracyjnego z zaznaczonymi czterema punktami o znanych współrzędnych. Następnie punkty te należy rozpoznać na obrazie kamery i podać ich współrzędne ekranowe, które za pomocą wzoru (3.4) pozwolą wyliczyć wektor WSP, a co za tym idzie wszystkie współczynniki szukanej transformacji.

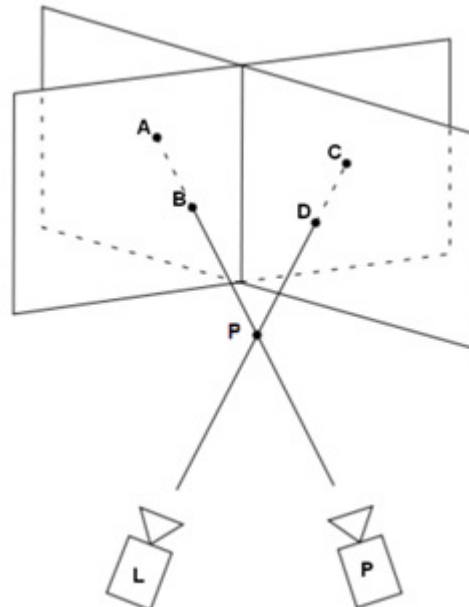
Do poprawnego odczytu pozycji w przestrzeni przez system stereowizyjny wymagany jest globalny układ współrzędnych oraz dwie płaszczyzny ze zdefiniowanymi układami współrzędnych o znanym położeniu względem układu globalnego. W pracy wykorzystano w tym celu wzorzec kalibracyjny. Wzorzec dysponuje dwoma płaszczyznami z zaznaczonymi punktami o znanych współrzędnych. Dodatkowo w rogu wzorca umieszczono początek globalnego układu współrzędnych. Rysunek 3.6 przedstawia widok z jednej z kamer przedstawiający wzorzec kalibracyjny z zaznaczonym początkiem globalnego układu współrzędnych oraz punktami, dzięki którym otrzymano transformacje płaskie.



Rysunek 3.6 - Wzorzec kalibracyjny z globalnym układem współrzędnych i punktami pozwalającymi na uzyskanie transformacji płaskich

Po obliczeniu wszystkich czterech transformacji płaski – po 2 płaszczyzny na każdą z kamer – można przystąpić do obliczenia trójwymiarowych współrzędnych punktu widocznego na obu kamerach. Algorytm działa według następujących kroków:

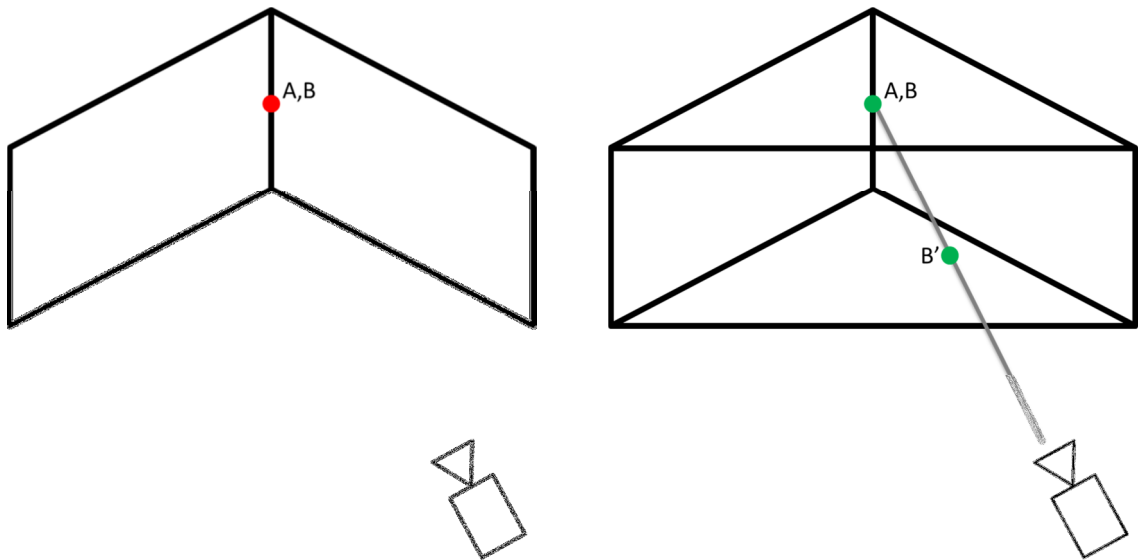
- 1) Odczytanie z obu kamer współrzędnych monitorowych szukanego punktu P
- 2) Obliczenie za pomocą transformacji płaskiej punktów na płaszczyznach kalibracyjnych (punkty A,B dla prawej kamery oraz C,D dla lewej)
- 3) Poprowadzenie prostych przez pary uzyskanych punktów
- 4) Odczytanie współrzędnych punktu ich przecięcia, czyli szukanego punktu P



*Rysunek 3.7 - Zasada działania stereowizji
Źródło: Zastosowanie stereowizji do odtwarzania trajektorii obiektów w przestrzeni 3D [5]*

Algorytm ten może napotkać na trudności, jeżeli obserwowany punkt leży na jednej linii z przecięciem płaszczyzn kalibracyjnych. W takim przypadku para punktów (na przykład A i B) będą stanowiły jeden punkt. Przeprowadzenie prostej przez jeden punkt jest możliwe na nieskończenie wiele sposobów, co uniemożliwia otrzymanie poprawnego odczytu. W praktyce, jeżeli punkty te znajdują się nie tylko w jednym punkcie, ale także w niewielkiej odległości od siebie algorytm ten uzyskuje bardzo duże błędy.

Rozwiązaniem tego problemu zastosowanym w pracy było stworzenie dodatkowej płaszczyzny pomocniczej, która pomaga uzyskać prostą w przypadku zbyt małej odległości między parą punktów uzyskanych z domyślnych transformacji płaskich. Płaszczyzna pomocnicza została stworzona ze skrajnych punktów wybieranych na wzorcu kalibracyjnym. W ten sposób punkt otrzymany z transformacji płaskiej na tej płaszczyźnie będzie zawsze w dużej odległości od punktów A i B, gdy te znajdują się blisko siebie.

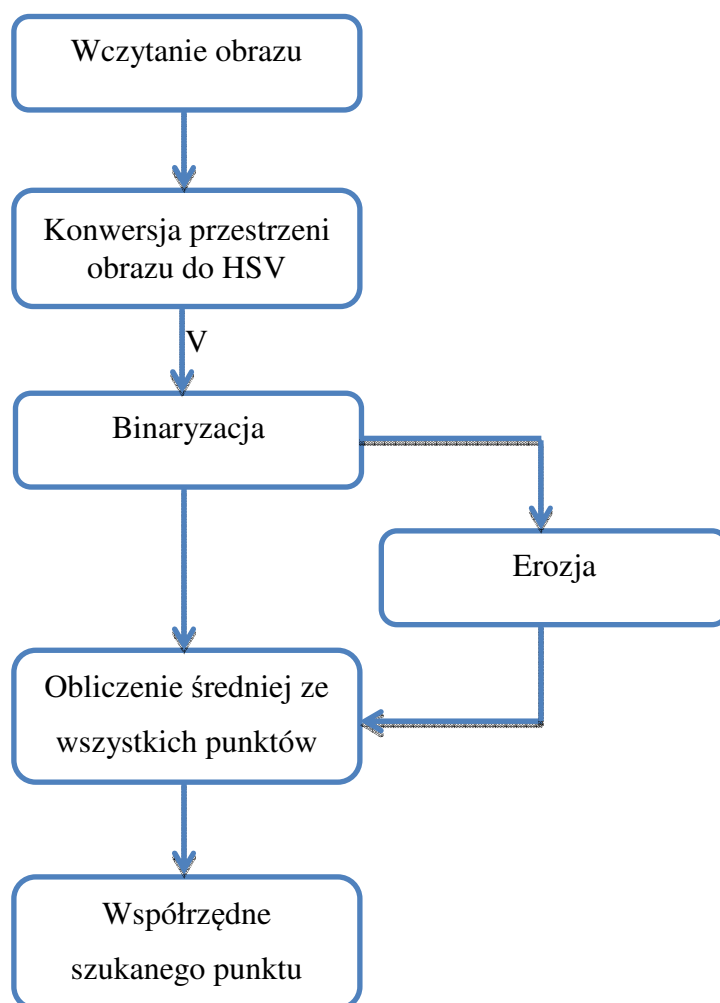


Rysunek 3.8 - A i B stanowią jeden punkt. Po lewej: Nie da się otrzymać prostej. Po prawej: dzięki wprowadzeniu płaszczyzny pomocniczej uzyskano punkt B' i dzięki niemu możliwe było poprowadzenie prostej.

3.4 Obserwacja położenia

Jak wspomniano do jednoznacznego określenia położenia helikoptera potrzebne są trzy współrzędne. Nie określają one jednak całej jednostki, a jeden konkretny punkt. Punkt ten za każdym razem musi zostać wykryty na obrazie – jego wybór ma zasadniczy wpływ na poziom skomplikowania algorytmu. W pracy wykorzystano zamontowaną w śmigłowcu diodę – zmieniono jedynie jej umiejscowienie. Diodę zamontowano pod helikopterem na osi obrotu śmigieł – jest to dobrze widoczny oraz stabilny punkt podczas lotu.

Przy poszukiwaniu na obrazie punktowego źródła światła, jakim jest dioda, nie ma potrzeby wykorzystywania skomplikowanych algorytmów obróbki obrazu. Już kilka prostych transformacji pozwala na jednoznaczne jej wykrycie. Wynika to z faktu, że jasne piksele mają wysokie wartości składowej Value w przestrzeni HSV. Algorytm użyty w pracy można prześledzić na poniższym schemacie.



Rysunek 3.9 - Schemat algorytmu wykrywania diody

Kamery użyte w pracy domyślnie przesyłają obraz w przestrzeni RGB, tak więc pierwszym z kroków algorytmu obróbki obrazu jest przekonwertowanie danych do przestrzeni HSV. Z punktu widzenia użytkownika nie zachodzą żadne zmiany – obraz w obu przestrzeniach wygląda dokładnie tak samo.

Kolejnym krokiem jest wydobycie z obrazu interesującej nas składowej. Jak wspomniano źródło światła najłatwiej jest wykryć poprzez wysokie wartości składowej Value. Obraz po pominięciu pozostałych dwóch wartości każdego z pikseli można opisać za pomocą dwuwymiarowej tablicy. Wizualnie obraz posiadający dla pojedynczego piksela zaledwie jedną wartość jest monochromatyczny. Wysokie wartości oznaczone są kolorem biały, a niskie, czarnym.

Następnie należy odizolować piksele o wysokich wartościach od tych ciemniejszych. W tym celu zastosowano binaryzację. Próg podziału wybrano analizując wartości, jakie przybierają piksele na zdjęciach z widoczną diodą. Stwierdzono, że na wszystkich ujęciach światło jest na tyle mocne, że wartości tworzące obraz źródła światła mają wartości bliskie możliwemu maksimum, czyli wartości 255. Niektóre z pikseli diody są ciemniejsze, jednak ich wartości nigdy nie są mniejsze niż 250. W algorytmie próg można ustawiać przed każdą próbą – wyniki testów przedstawiono w dalszej części pracy.

Po binaryzacji otrzymano obraz z wartościami 1 oraz 0. Jednak nie uzyskano pojedynczego, wyróżniającego się punktu, który można by uznać za źródło światła. Zauważono też, że obiektem na obrazie w wielu przypadkach jest nie tylko dioda, ale także niewielkie elementy, które powstają na skutek zbytniego oświetlenia obserwowanej sceny oraz w wyniku odbicia światła od otoczenia helikoptera. Aby z otrzymanego w wyniku binaryzacji obiektu uzyskać punkt należy znaleźć jego środek ciężkości poprzez uśrednienie wszystkich jego punktów. Ten sposób ma jednak swoją wadę – zbyt duża ilość punktów obiektu niebędących diodą zakłóca poprawne uzyskanie współrzędnych szukanego punktu. Proszym rozwiązaniem jest przeprowadzenie na obrazie zabiegu erozji. W jej wyniku małe obiekty znikną, a większe zmaleją nie zniekształcając swojego środka ciężkości, którego szukamy. Rozwiązanie to też nie jest idealne. Działa ono tylko w przypadku, gdy widziana dioda jest większa niż zakłócenia. Problemy pojawiają się, gdy śmigłowiec znajduje się

daleko od kamery. W takim przypadku obiekt przed erozją może być tak mały, że po jej przeprowadzeniu obiekt zniknie całkowicie. Algorytm sprawdza wynik erozji – jeżeli istnieje obiekt na obrazie wynikowym to jego środek ciężkości brany jest pod uwagę. W przeciwnym wypadku analizowany jest obraz sprzed erozji. Sprawdzono działanie algorytmu dla różnych wielkości elementu strukturalnego dla operacji erozji, a wyniki omówiono w dalszej części. Po uśrednieniu współrzędnych wszystkich punktów z wybranego obrazu otrzymano współrzędne monitorowe diody.

Dokonano także próby ulepszenia stworzonego algorytmu. W momencie przeprowadzania binaryzacji niezbędny jest próg podziału pikseli. Jego optymalna wartość zależy bezpośrednio od otoczenia. Jakikolwiek zbyt jasne jego elementy zostaną błędnie zakwalifikowane jako obiekt, a w konsekwencji mogą znacznie przesunąć punkt wykrycia diody. Zaproponowanym rozwiązaniem było zapamiętanie obrazu początkowego, bez helikoptera, a następnie odejmowanie go od obrazu otrzymywanego z kamery. Rozwiązanie to powinno zniwelować wpływ otoczenia na wyniki uzyskiwane w wyniku obróbki obrazu. Po odejmowaniu nie wiadomo, jaką wartość będzie mieć dioda, więc nie można ustawić sztywno progu binaryzacji – w algorytmie ustawia się próg jako procent najjaśniejszego piksela.

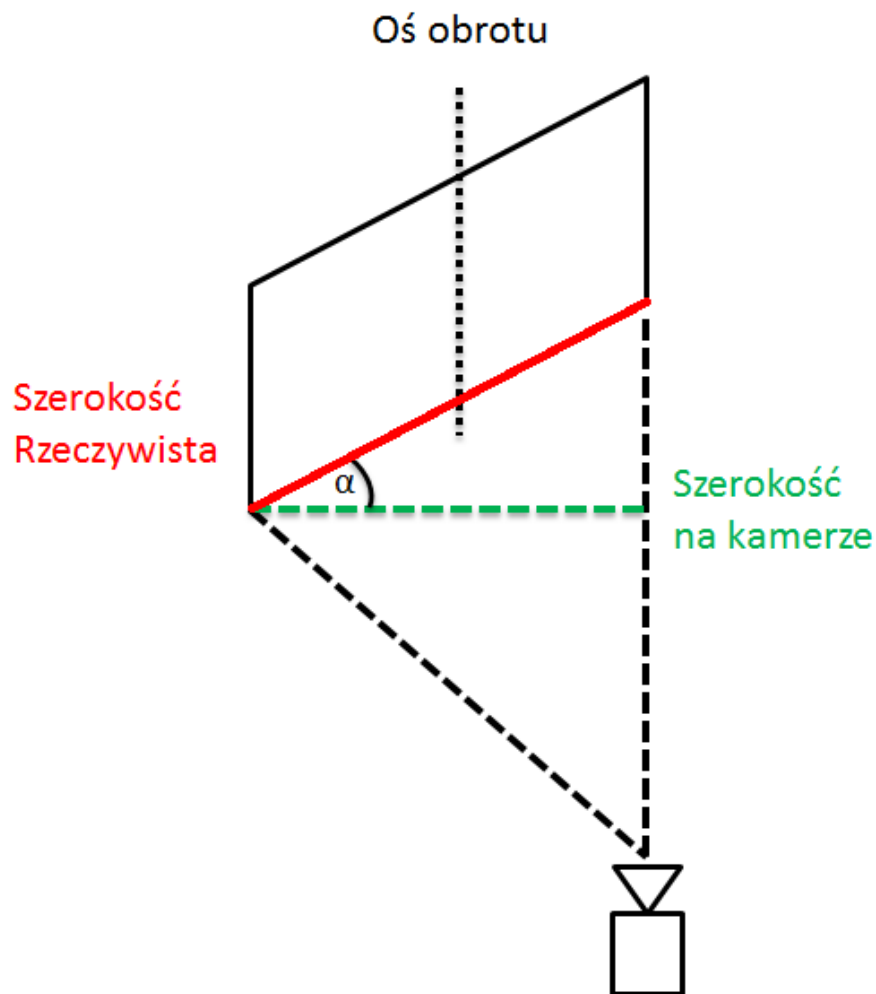
Niestety rozwiązanie to nie jest idealne. Występuje w nim kilka problemów, których nie doświadczono w poprzedniej wersji algorytmu. Po pierwsze odejmowanie nie sprawdza się w przypadku, gdy otoczenie nie jest stałe. Szczególnie dotyczy to oświetlenia. Nawet nieznacznie poruszające się jasne elementy tworzą po odjęciu maksima lokalne, które często mają wartości wyższe niż piksele tworzące diodę. Efekt taki wcale nie jest trudny do uzyskania – w trakcie prób przeprowadzanych w trakcie realizacji pracy zdarzył się niejednokrotnie. Słoneczny dzień w połączeniu z lekko poruszającymi się wskutek przeciągu żaluzjami spowodował pojawienie się na obrazie kamery jasnych pasów nieznacznie zmieniających swoje położenie. Kolejny problem pojawia się, gdy helikopter podczas lotu zasłania jasne piksele. Wówczas różnica w jasności między świecąca dioda a otoczeniem jest niewielka – po binaryzacji piksele takie mogą stać się tłem, a nie obiektem.

Oba algorytmy, z różnymi wartościami parametrów zostały porównane na kilku nagranych lotach w następnym rozdziale.

3.5 Obserwacja orientacji

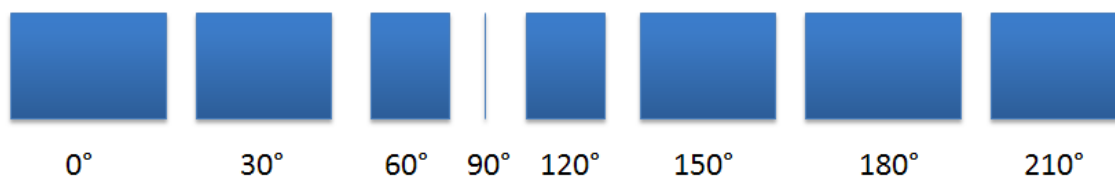
W poprzednim podrozdziale opisano zasadę pozyskiwania z obrazu informacji o położeniu helikoptera. Jak wspomniano wcześniej do pełnego opisu jednostki brakuje jeszcze wiedzy o jego orientacji, czyli w analizowanym przypadku o kącie obrotu wokół osi pionowej.

Pomysł wykorzystany w pracy bazował na fakcie, że przy obrocie prostokąta wokół pionowej osi będącą osią jego symetrii, obserwator stojący obok zawsze będzie widział prostokąt o tej samej wysokości – zmienna będzie tylko szerokość figury. Obserwator ten po zmierzeniu szerokości widzianego przez siebie prostokąta i porównaniu jej z rzeczywistą szerokością figury otrzyma kąt obrotu prostokąta. Poniższy rysunek obrazuje objaśniane zagadnienie.



Rysunek 3.10 - Metoda pozyskiwania kąta obrotu prostokąta

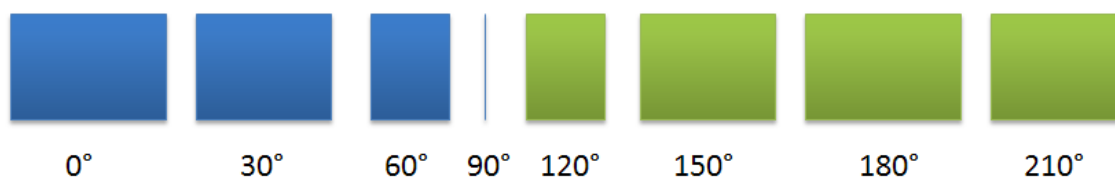
Poddano obrót prostokąta dokładniejszej analizie. W trakcie obrotu figury występują pewne sytuacje, w których obserwator widzi dokładnie ten sam obraz, mimo iż kąt obrotu jest inny.



Rysunek 3.11 - Obrót prostokąta widziany przez obserwatora

Sytuacja ta występuje dla miar kąta symetrycznych względem 90 oraz 270 stopni, a także dla kątów, których różnica wynosi 180 stopni. Oznacza to, że widząc pewien prostokąt na obrazie kamery nie można jednoznacznie stwierdzić pod jakim kątem obrócona jest figura – należy dokonać modyfikacji.

W celu odróżnienia kątów symetrycznych względem 90 i 270 stopni należy oznaczyć drugą stronę prostokąta. W pracy wykorzystano zmianę koloru.



Rysunek 3.12 - Obrót prostokąta widziany przez operatora po zmianie koloru drugiej strony prostokąta

Śmigłowiec nie posiada jednak żadnych prostokątów, na podstawie których można by przeprowadzić powyższe rozumowanie. W tym celu stworzono dwa markery w kształcie prostokąta o znanych wymiarach. Dla lepszej dokładności systemu wizyjnego wszelkie rozpoznawane obiekty powinny być jak największe, jednak niewielkie rozmiary jednostki pozwoliły na stworzenie prostokątów o wymiarach 1cm x 3cm. Prostokąty te powinny być łatwo rozpoznawalne mimo różnych kolorów – zdecydowano się na niebieską i zieloną barwę. Kolory te mają bardzo małą ilość składowej czerwonej, co w połączeniu z zastosowaną białą obwódką wokół markerów pozwala na łatwe wykrycie poprzez szukanie niskich wartości R w przestrzeni RGB.

Ponizej przedstawiono zdjęcia śmigłowca po zamontowaniu omawianych markerów:

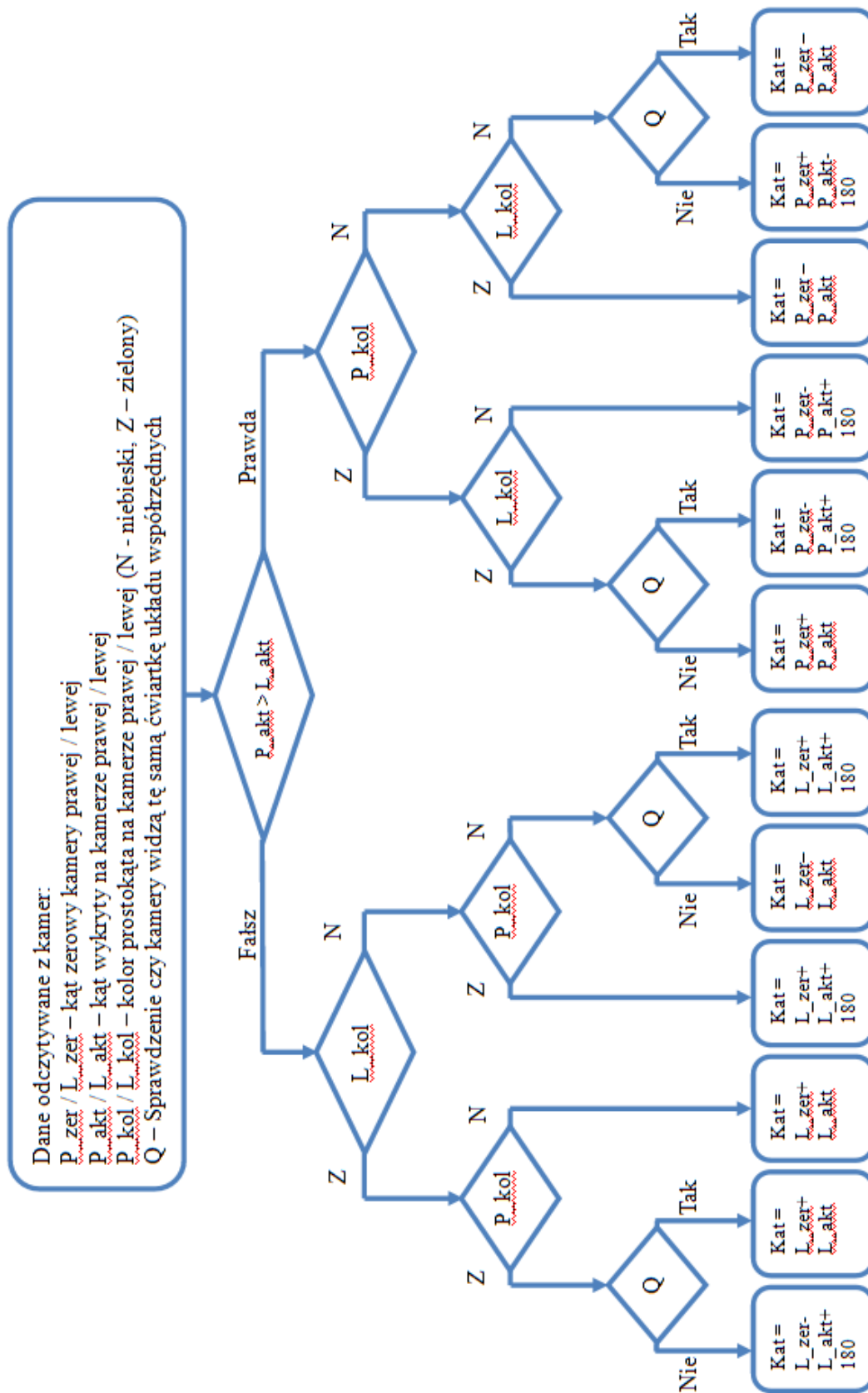


Rysunek 3.13 - Helikopter z zamontowanymi markerami (widok z obu stron)

Po takiej modyfikacji rozwiązano jeden z problemów, jednak drugi wciąż pozostał – prostokąt dla obrotu równego 150° i 210° wygląda dokładnie tak samo. Problem ten pojawia się, gdy jedna z kamer przechodzi w okolicach normalnej do prostokąta – otrzymujemy wtedy mamy ten sam kąt na kamerze (bliski 90°) a inną w rzeczywistości orientacje helikoptera. W celu odróżnienia takich dwóch sytuacji porównywana jest różnica między kątami na kamerach. Nie może być ona jednak porównywana wprost (przez równość), ponieważ pomiary są niedokładne - wprowadzono więc nierówność. Jeżeli różnica będzie większa od pierwotnej to jedynym powodem może być niedokładnością pomiarów. Jeżeli będzie mniejsza to albo znowu spowodowane jest to niedokładnością pomiaru albo jedna z kamer znajduje się w innej ćwiartce układu współrzędnych - założono że mała odchyłka od początkowej różnica wynika z niedokładności a dopiero większa od zmiany ćwiartki (dobór progu jest kluczowy). Rozumowanie to wymaga informacji o różnicy kątowej między kamerami. W tym celu przed wszelkimi pomiarami należy umieścić helikopter w polu widzenia obu kamer w taki sposób, aby obie znajdowały się w tej samej ćwiartce układu współrzędnych niebieskiego prostokąta i po analizie obrazu obliczyć różnicę kątową. Dodatkowo kąt, pod jakim ustawiony jest w tym momencie śmigłowiec traktowany jest jako zerowy kąt obrotu jednostki.

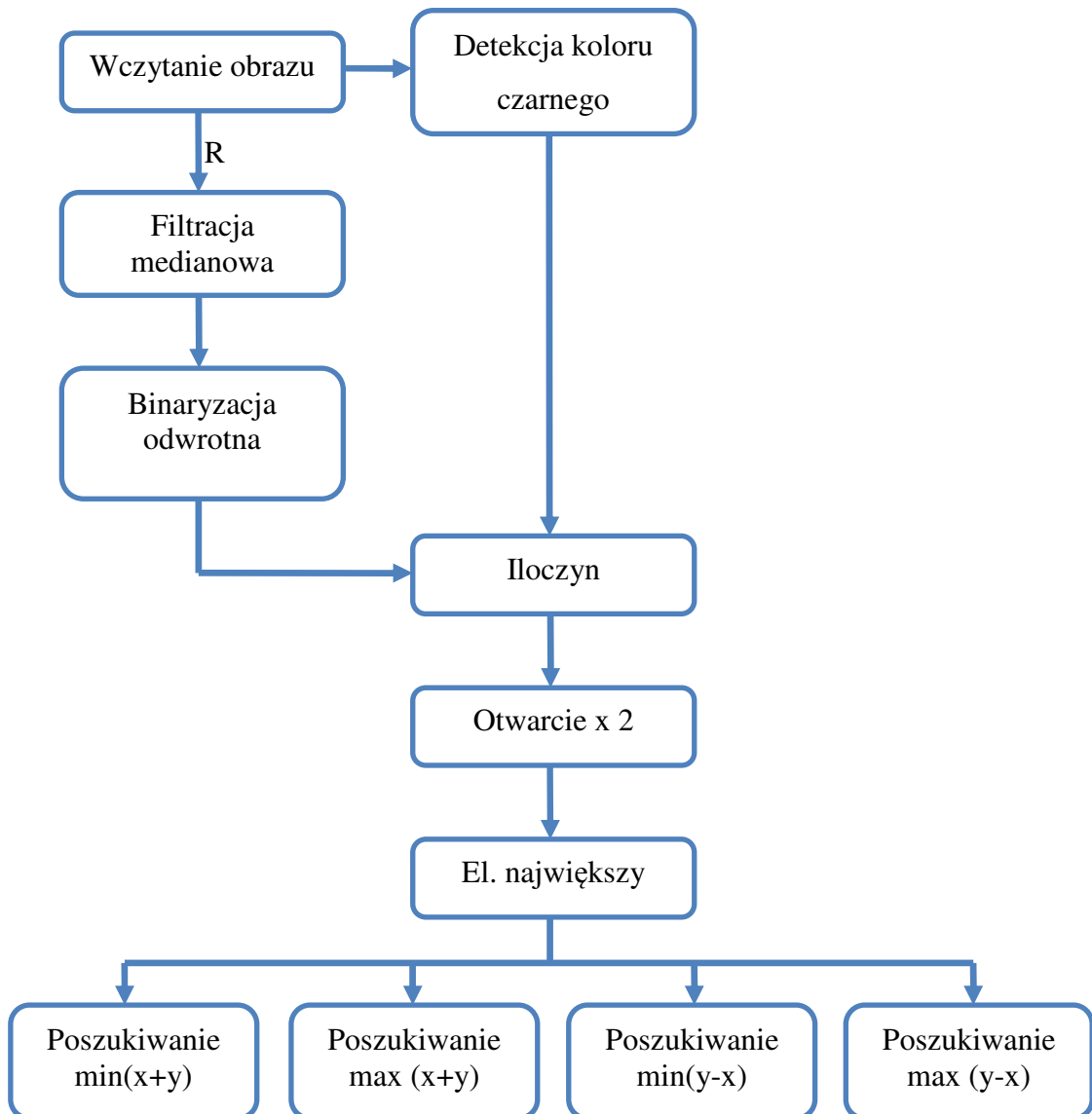
Zauważono także jeszcze jedną niedogodność – szerokość widzianego na kamerze prostokąta zależy także od odległości helikoptera od obiektywu. Rozwiązanie jest jednak bardzo proste – szerokość zależy od odległości tak samo jak wysokość. Oznacza to, że ich stosunek jest stały, niezależny od aktualnego położenia jednostki. Ostatecznie algorytm stworzony na potrzeby pracy porównuje właśnie wspomniany stosunek długości boków figury, a nie jej samą szerokość.

Ostatecznie algorytm można przedstawić za pomocą poniższego schematu blokowego:



Rysunek 3.14 – Algorytm pozyskiwania kąta obrotu helikoptera na podstawie kąta widzianego z obu kamer

Przedstawiony algorytm oblicza orientację helikoptera na podstawie informacji o długościach boków oraz koloru prostokątów widocznych na obrazie z obu kamer. Należało stworzyć kolejny algorytm, który wykrywałby szukane wartości na obrazie. Długości boków można w łatwy sposób otrzymać znając współrzędne wierzchołków figury. Kolor można łatwo sprawdzić porównując wartości odpowiednich składowych dla środka znalezionej prostokąta. Zastosowaną w pracy obróbkę obrazu można przedstawić za pomocą poniższego schematu blokowego.



Rysunek 3.15–Algorytm wyszukiwania prostokąta

Algorytm korzysta z faktu, że stworzone na potrzeby pracy markery posiadają kolory o bardzo niewielkiej ilości składowej czerwonej. Po wczytaniu obrazu wybierana jest właśnie ta składowa i zostaje ona poddana filtracji medianowej. Zabieg ten ma na celu zmniejszenie wpływu zakłóceń na pomiar oraz ujednoczenia wartości na szukanym prostokącie. Następnie obraz ten poddawany jest binaryzacji. Przyporządkowanie jest tu jednak odwrotne – wartość 1 otrzymują piksele o dostatecznie niskiej wartości. Próg transformacji został dobrany eksperymentalnie, a wyniki dla jego różnych wartości zestawiono w dalszej części pracy. Po takich zabiegach otrzymano prostokąt oraz inne elementy o niewielkiej wartości składowej czerwonej. W większości jest to kolor czarny występujący na helikopterze.

Dla obrazu w takiej formie nie da się w prosty sposób jednoznacznie stwierdzić, jakie współrzędne mają wierzchołki szukanej figury. Należało usunąć z obrazu elementy powstałe z koloru czarnego. W tym celu wykonano na obrazie wejściowym detekcję tegoż właśnie koloru – obraz wynikowy po detekcji posiada wartości 0 dla pikseli o wartościach wszystkich trzech składowych poniżej zadanego progu.

W wyniku iloczynu obu obrazów otrzymujemy obraz, na którym obiekt stanowi szukany prostokąt oraz bardzo małe elementy powstałe z zakłóceń oraz niedokładności detekcji koloru czarnego. Następnie przeprowadzono dwukrotnie operacje otwarcia. Mają one na celu zapobiegnięcie ewentualnemu połączeniu się prostokąta z innymi elementami otoczenia. Zaraz po tych operacjach do dalszej obróbki wybierany jest element największy obrazu.

Ostatnia operacją jest maksymalizacja i minimalizacja sumy oraz różnicy współrzędnych. Taki zabieg pozwala na uzyskanie współrzędnych wierzchołków szukanej figury. Korzystając z twierdzenia pitagorasa obliczono na ich podstawie długości boków prostokąta, a następnie ich stosunek, który posłuży do wyznaczenia orientacji helikoptera według wcześniej omawianego schematu.

Sprawdzenie poprawności omówionego algorytmu przeprowadzono w następnym rozdziale.

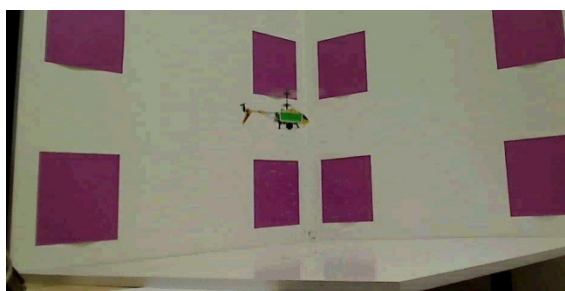
4. Testy

4.1 Test algorytmu wykrywania diody

W trakcie tworzenia pracy nagrano kilka lotów helikoptera. Filmy te różnią się warunkami świetlnymi otoczenia, a także ustawieniem kamer. Wszystkie filmy podzielono na klatki, a następnie poddano je obróbce zaproponowanej w rozdziale 3.4. Wyliczone w ten sposób punkty naniesiono na odpowiednie klatki i stworzono z nich film. W ten sposób można zobaczyć oryginalny film z zaznaczonym punktem wykrycia diody.

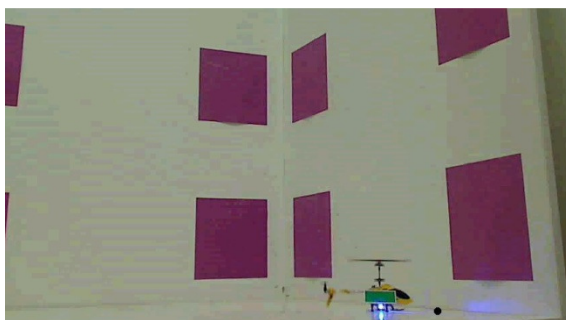
Kolejnym krokiem było stwierdzenie, czy algorytm zadziałał poprawnie. W tym celu każdą z klatek wynikowego filmu przydzielono do jednej z trzech kategorii :

- Poprawnie wykryta – oznacza to, że czarny punkt naniesiony na film wynikowy zakrywa świecąca diodę.



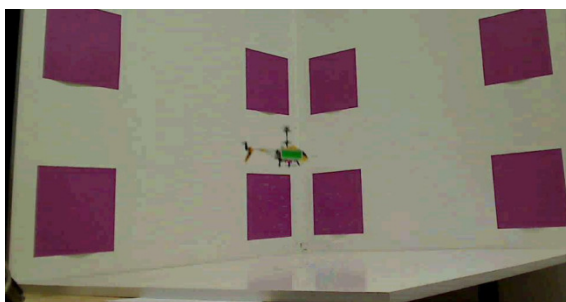
Rysunek 4.1 - Klatka filmu z poprawnie wykrytą diodą

- Niepoprawnie wykryta z winy otoczenia – oznacz to, że czarny punkt nie pokrywa diody, ale znajduje się na innym źródle światła (np. odbitego), bądź między diodą, a innym źródłem światła.



Rysunek 4.2 - Klatka filmu z niepoprawnie wykrytą diodą z powodu widocznego odbicia światła od otoczenia

- Niepoprawnie wykryta z winy algorytmu – oznacza to, że czarny punkt nie pokrywa diody, ani żadnego innego źródła światła. Często punkt ten znajduje się w skrajnym rogu klatki filmu o współrzędnych (0,0)



Rysunek 4.3 - Klatka filmu z niepoprawnie wykrytą diodą z powodu źle dobranych parametrów algorytmu – brak czarnego punktu

Wszystkie filmy poddano obróbce z użyciem różnych wartości poszczególnych parametrów, czyli progu binaryzacji i wielkości elementu strukturalnego przy erozji. W pracy użyto elementu strukturalnego w postaci kwadratu, co oznacza, że przy ustawieniu jego wielkości na 1 erozja nie występuje. Na płycie CD dołączonej do pracy można znaleźć wszystkie filmy – zarówno te oryginalne, jak i te po obróbce.

Na kolejnej stronie zestawiono wyniki analizy pierwszego z nagranych filmów:

Film 1 – kamera lewa – 29 klatek

Nazwa filmu	Próg Binarizacji	Wielkość elementu strukturalnego	Liczba klatek poprawnych (%)	Liczba klatek złych z powodu otoczenia (%)	Liczba klatek złych (%)
Film1L_240_1	240	1	27 (94%)	1 (3%)	1 (3%)
Film1L_240_3	240	3	24 (83%)	4 (14%)	1(3%)
Film1L_240_4	240	5	26 (90%)	2 (7%)	1 (3%)
Film1L_250_1	250	1	25 (86%)	2 (7%)	2 (7%)
Film1L_250_3	250	3	21 (73%)	6 (20%)	2 (7%)
Film1L_250_5	250	5	26 (90%)	1 (3%)	2 (7%)
Film1L_255_1	255	1	25 (86%)	2 (7%)	2 (7%)
Film1L_255_3	255	3	25 (86%)	2 (7%)	2 (7%)
Film1L_255_5	255	5	25 (86%)	2 (7%)	2 (7%)

Film 1 – kamera prawa – 29 klatek

Nazwa filmu	Próg Binarizacji	Wielkość elementu strukturalnego	Liczba klatek poprawnych (%)	Liczba klatek złych z powodu otoczenia (%)	Liczba klatek złych (%)
Film1P_240_1	240	1	28 (97%)	1 (3%)	0 (0%)
Film1P_240_3	240	3	25 (86%)	4 (14%)	0 (0%)
Film1P_240_4	240	5	29 (100%)	0 (0%)	0 (0%)
Film1P_250_1	250	1	29 (100%)	0 (0%)	0 (0%)
Film1P_250_3	250	3	27 (93%)	2 (7%)	0 (0%)
Film1P_250_5	250	5	29 (100%)	0 (0%)	0 (0%)
Film1P_255_1	255	1	27 (93%)	2 (7%)	0 (0%)
Film1P_255_3	255	3	25 (86%)	4 (14%)	0 (0%)
Film1P_255_5	255	5	28 (97%)	1 (3%)	0 (0%)

Błędne wykrycia wynikają z częstego odbijania się światła od powierzchni, na której ląduje helikopter oraz ściany bocznej, na której widać duże odbicie pod koniec filmu. Jednak niektóre z wersji algorytmu były w stanie odrzucić te refleksy – najprawdopodobniej w trakcie binaryzacji. Na większości nawet tych dobrze rozpoznanych klatek filmu widać niewielkie przesunięcie wykrytego punktu od diody. Powodem tego jest wspomniane już odbicie światła diody od podłoża. Całkowicie błędnie rozpoznane klatki są spowodowane zbyt ciemnymi pikselami diody przy szybkim ruchu jednostki (obraz jest rozmyty). W takim przypadku po binaryzacji piksele te zaliczane są do tła.

Następnie dokonano analizy kolejnego filmu nagranych w innych warunkach świetlnych z innym ustawieniem kamer. Wyniki zestawiono poniżej.

Film 2 – kamera lewa – 35 klatek

Nazwa filmu	Próg Binarizacji	Wielkość elementu strukturalnego	Liczba klatek poprawnych (%)	Liczba klatek złych z powodu otoczenia (%)	Liczba klatek złych (%)
Film1L_240_1	240	1	32 (94%)	3 (9%)	0 (0%)
Film1L_240_3	240	3	33 (94%)	2 (6%)	0 (0%)
Film1L_240_4	240	5	33 (94%)	2 (6%)	0 (0%)
Film1L_250_1	250	1	34 (97%)	1 (3%)	0 (0%)
Film1L_250_3	250	3	33 (94%)	2 (6%)	0 (0%)
Film1L_250_5	250	5	34 (97%)	1 (3%)	0 (0%)
Film1L_255_1	255	1	34 (97%)	1 (3%)	0 (0%)
Film1L_255_3	255	3	34 (97%)	1 (3%)	0 (0%)
Film1L_255_5	255	5	34 (97%)	1 (3%)	0 (0%)

Film 2 – kamera prawa – 35 klatek

Nazwa filmu	Próg Binaryzacji	Wielkość elementu strukturalnego	Liczba klatek poprawnych (%)	Liczba klatek złych z powodu otoczenia (%)	Liczba klatek złych (%)
Film1P_240_1	240	1	25 (72%)	10 (28%)	0 (0%)
Film1P_240_3	240	3	27 (77%)	8 (23%)	0 (0%)
Film1P_240_4	240	5	32 (91%)	3 (9%)	0 (0%)
Film1P_250_1	250	1	29 (83%)	6 (17%)	0 (0%)
Film1P_250_3	250	3	32 (91%)	3 (9%)	0 (0%)
Film1P_250_5	250	5	33 (94%)	2 (6%)	0 (0%)
Film1P_255_1	255	1	31 (89%)	4 (11%)	0 (0%)
Film1P_255_3	255	3	34 (97%)	1 (3%)	0 (0%)
Film1P_255_5	255	5	33 (94%)	2 (6%)	0 (0%)

Film ten jest trudniejszy do obróbki w porównaniu do poprzedniego. Helikopter wzbijając się ustawiony jest dziobem w kierunku kamery, przez co zasłania źródło światła – widać tylko odbicie od podłoża. Na końcu filmu helikopter ląduje blisko ściany bocznej powodując duży refleks świetlny, który w miarę zwiększania progu binaryzacji jest coraz lepiej niwelowany przez algorytm. Z drugiej strony źródło światła nigdy nie jest przesłonięte płożą, co zaowocowało brakiem źle wykrytych klatek.

Podsumowując najlepsze wyniki otrzymano dla algorytmu o progu binaryzacji równym 250 oraz elemencie strukturalnym o wielkości 5. Algorytm ten we wszystkich 4 nagraniach wykrył poprawnie 122 z 128 klatek (95%).

4.2 Test algorytmu wykrywania diody z odejmowaniem tła

Kolejnym krokiem było nagrań użytych w poprzednim podrozdziale obróbce z wykorzystaniem wcześniej zapamiętanego tła. W przypadku tego algorytmu ustawiano te same parametry, czyli próg binaryzacji i wielkość erozji. Należy jednak pamiętać, że próg binaryzacji dla tej wersji algorytmu jest ustawiany jako procentowa wartość najjaśniejszego piksela.

Filmy poddano obróbce i zanalizowano według takich samymi kryteriów, jak w poprzednim podrozdziale. Wszystkie filmy zostały zapisane i zawarte na płycie CD dołączonej do pracy.

Film 1 – kamera lewa – 29 klatek

Nazwa filmu	Próg Binarizacji	Wielkość elementu strukturalnego	Liczba klatek poprawnych (%)	Liczba klatek złych z powodu otoczenia (%)	Liczba klatek złych (%)
Film1L_090_1	0,9	1	14 (48,5%)	1 (3%)	14 (48,5%)
Film1L_090_3	0,9	3	14 (48,5%)	1 (3%)	14 (48,5%)
Film1L_090_5	0,9	5	14 (48,5%)	1 (3%)	14 (48,5%)
Film1L_095_1	0,95	1	16 (55%)	1 (3%)	12 (42%)
Film1L_095_3	0,95	3	16 (55%)	1 (3%)	12 (42%)
Film1L_095_5	0,95	5	16 (55%)	1 (3%)	12 (42%)
Film1L_099_1	0,99	1	17 (59%)	1 (3%)	11 (38%)
Film1L_099_3	0,99	3	17 (59%)	1 (3%)	11 (38%)
Film1L_099_5	0,99	5	17 (59%)	1 (3%)	11 (38%)

Film 1 – kamera prawa – 29 klatek

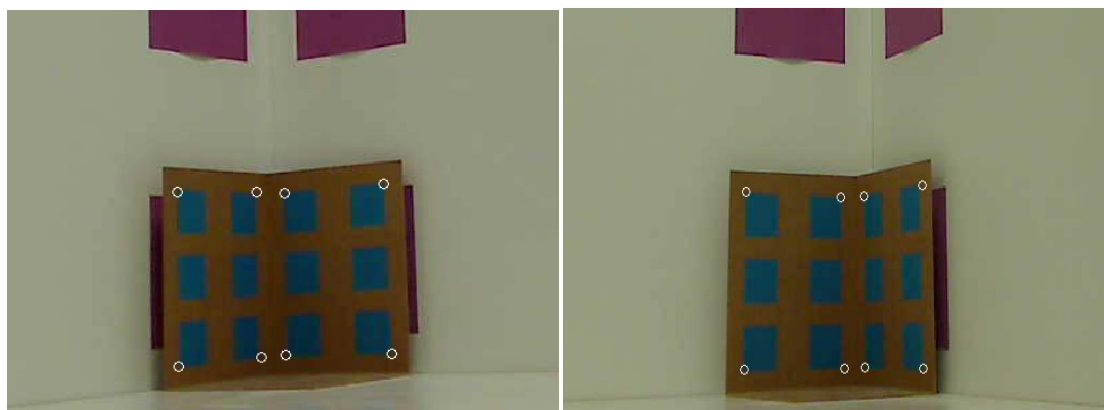
Nazwa filmu	Próg Binaryzacji	Wielkość elementu strukturalnego	Liczba klatek poprawnych (%)	Liczba klatek złych z powodu otoczenia (%)	Liczba klatek złych (%)
Film1P_090_1	240	1	21 (73%)	1 (3%)	7 (24%)
Film1P_090_3	240	3	23 (79%)	2 (7%)	4 (14%)
Film1P_090_5	240	5	22 (76%)	0 (0%)	7 (24%)
Film1P_095_1	250	1	21 (73%)	1 (3%)	7 (24%)
Film1P_095_3	250	3	22 (76%)	1 (3%)	6 (21%)
Film1P_095_5	250	5	21 (73%)	1 (3%)	7 (24%)
Film1P_099_1	255	1	22 (76%)	0 (0%)	7 (24%)
Film1P_099_3	255	3	22 (76%)	0 (0%)	7 (24%)
Film1P_099_5	255	5	22 (76%)	0 (0%)	7 (24%)

Już po analizie pierwszego filmu można stwierdzić, że wyniki są gorsze niż w przypadku algorytmu niewykorzystującego odejmowania obrazów. Zmniejszone są jednak ilość niepoprawnie rozpoznanych klatek z powodu odbicia światła – wynika to z faktu, że światło to odbijało się od białej powierzchni, co za tym idzie różnica wartości była niewielka.

Występuje jednak duża ilość klatek, na którym dioda wykryta jest w punkcie odległym od helikoptera. Źródłem takiego błędu jest zmieniające się oświetlenie sceny w porównaniu do odejmowanego od niej obrazu zerowego zarejestrowanego przed lotem. Po odjęciu powstają w ten sposób bardzo jasne punkty, które zostają błędnie rozpoznane, jako szukana dioda.

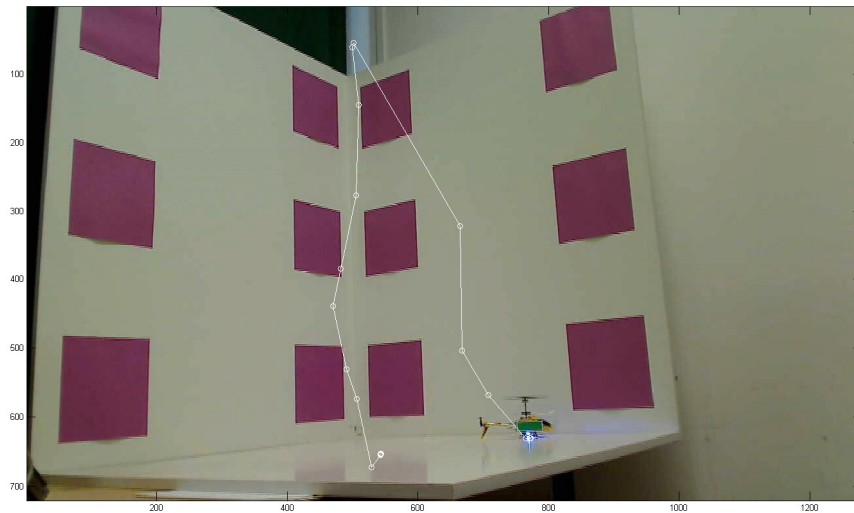
4.3 Test działania stereowizji

Kolejnym krokiem było sprawdzenie, czy ze współrzędnych poprawnie wykrytej diody system jest w stanie otrzymać jej poprawne współrzędne. W tym celu obliczono trajektorie, po jakiej poruszał się helikopter. Do obliczeń niezbędna jest kalibracja obu kamer.

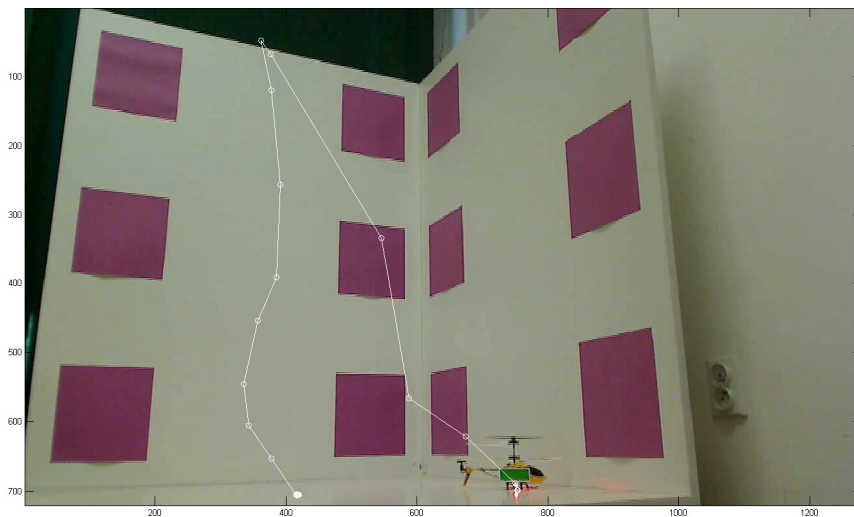


Rysunek 4.4 - Widok z kamery lewej i prawej na wzorec kalibracyjny z zaznaczeniem szukanych punktów

Nagrania z obu kamer poddano obróbce w celu znalezienia diody. W algorytmie wyszukiwania diody użyto progu binaryzacji o wartości 250 i erozji z elementem o wielkości 5. Te parametry uznano za najlepsze. Zebrano uzyskane punkty i zaznaczono je na ostatniej klatce filmu. Efekt widoczny poniżej.

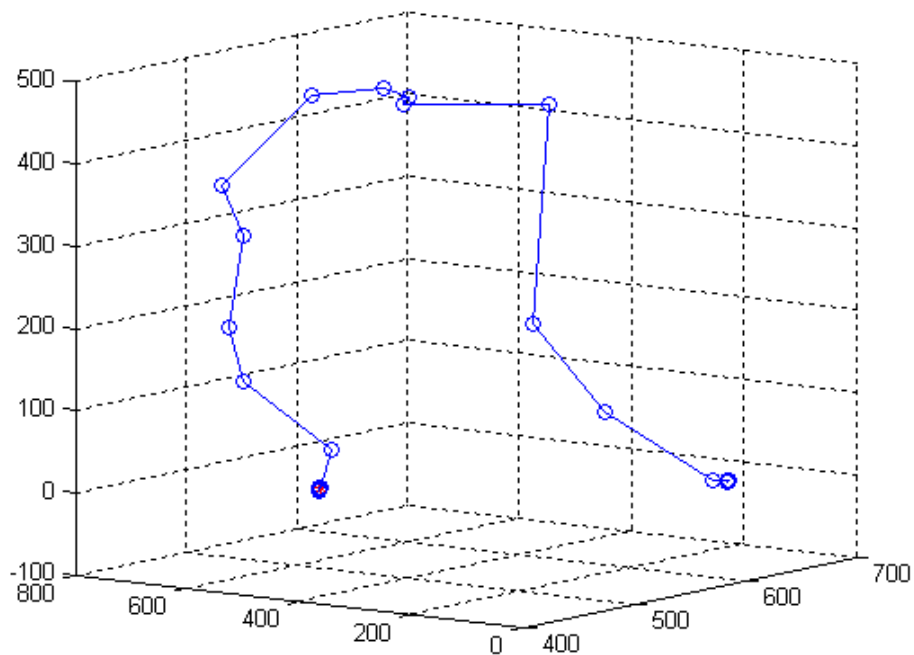


Rysunek 4.5- Widok z kamery lewej z zaznaczonymi położeniami diody



Rysunek 4.6 - Widok z kamery prawej z zaznaczonymi położeniami diody

Następnie dzięki omówionemu wcześniej algorytmowi stereowizyjnemu obliczono punkty, w których znajdował się helikopter, a następnie na ich podstawie wykreślono jego trajektorie.



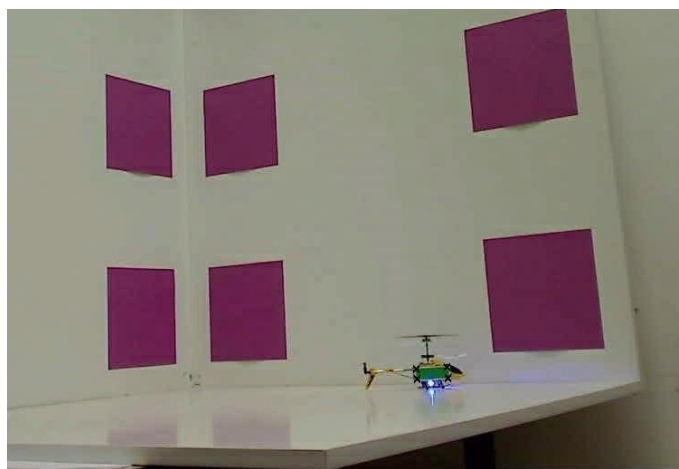
Rysunek 4.7 - Trajektoria helikoptera

Zauważono, że trajektoria zakreślana przez helikopter na filmach jest zbliżona do tej przedstawionej na wykresie trójwymiarowym. Pewne niedokładności wynikają oczywiście z rozbieżności między wykrytym punktem diody, a jej faktycznym położeniem oraz z niedokładnego wyboru punktów przy kalibracji. Wszystkie pliki z testu zostały zawarte na płycie CD.

4.4 Test algorytmu wykrywania prostokąta

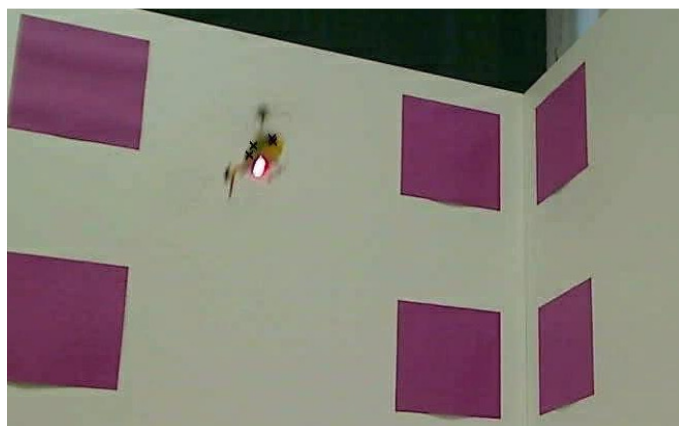
Kilka filmów nakręconych na potrzeby pracy poddano obróbce opisanej w podrozdziale 3.5. Otrzymano w ten sposób wynikowe filmy z zaznaczonymi rogami wykrytego prostokąta oraz orientacje helikoptera w kolejnych chwilach czasowych. Następnie filmy te poddano analizie wizualnej przypisując każdą z klatek do jednej z trzech poniższych grup:

- Poprawnie wykryta – czarne krzyżyki oznaczające wykryte rogi prostokąta znajdują się w pobliżu prostokąta widocznego na obrazie



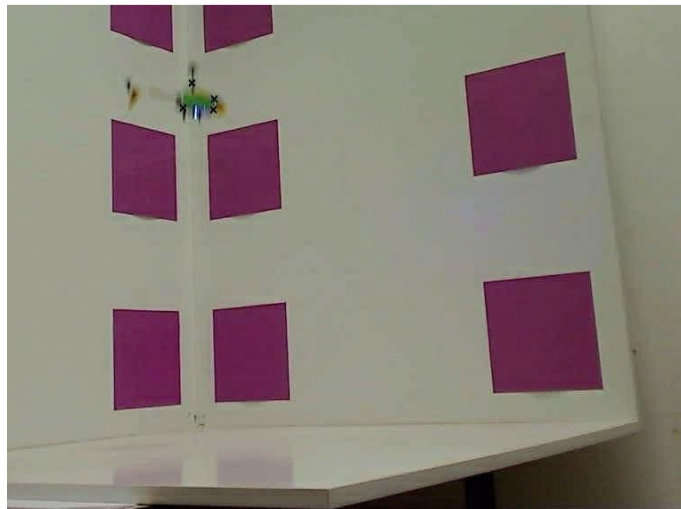
Rysunek 4.8 - Klatka filmu z poprawnie wykrytym prostokątem

- Niemożliwa do wykrycia – prostokąt nie jest widoczny przez kamerę, bądź rozmazanie klatki jest zbyt duże



Rysunek 4.9–Klatka filmu z kątem obrotu helikoptera uniemożliwiającym poprawne wykrycie

- Niepoprawnie wykryta z winy algorytmu – czarne krzyżyki znajdują się z dala od wykrywanego prostokąta, a jest on widoczny na obrazie



Rysunek 4.10–Klatka filmu z niepoprawnie wykrytym prostokątem – jeden z markerów został wykryty poza prostokątem

Wszystkie filmy poddano obróbce z użyciem różnych wartości poszczególnych parametrów, czyli progu wykrycia koloru czarnego, progu binaryzacji oraz wielkości elementów strukturalnych przy operacjach otwarcia. W pracy użyto elementu strukturalnego w postaci kwadratu, co oznacza, że przy ustawieniu jego wielkości na 1 erozja nie występuje. Na płycie CD dołączonej do pracy można znaleźć wszystkie filmy – zarówno te oryginalne, jak i te po obróbce.

Obróbce poddano ten sam film, który występuje w poprzednich podrozdziałach, czyli Film 1, oraz film dotychczas nieobrabiany oznaczony numerem 3

Na kolejnych stronach zestawiono wyniki analizy pierwszego z nagranych filmów:

Film 1 – kamera lewa – 29 klatek

Nazwa filmu	Próg koloru czarnego	Próg binaryzacji	Wielkość El. Strukturalnego nr 1 / nr 2	Liczba klatek poprawnych (%)	Liczba klatek niemożliwych (%)	Liczba klatek złych (%)
Film1L_40_50_3_1	40	50	3/1	16(56%)	4(14%)	9(30%)
Film1L_40_50_3_3	40	50	3/3	17(59%)	4(14%)	8(27%)
Film1L_40_50_5_1	40	50	5/1	17(59%)	4(14%)	8(27%)
Film1L_40_50_5_3	40	50	5/3	19 (64%)	4(14%)	6(20%)
Film1L_40_60_3_1	40	60	3/1	19 (64%)	4(14%)	6(20%)
Film1L_40_60_3_3	40	60	3/3	20 (69%)	4(14%)	5(17%)
Film1L_40_60_5_1	40	60	5/1	19 (64%)	4(14%)	6(20%)
Film1L_40_60_5_3	40	60	5/3	19 (64%)	4(14%)	6(20%)
Film1L_40_70_3_1	40	70	3/1	20 (69%)	4(14%)	5(17%)
Film1L_40_70_3_3	40	70	3/3	18(62%)	4(14%)	7(24%)
Film1L_40_70_5_1	40	70	5/1	19 (64%)	4(14%)	6(20%)
Film1L_40_70_5_3	40	70	5/3	20 (69%)	4(14%)	5(17%)
Film1L_60_50_3_1	60	50	3/1	21(72%)	4(14%)	4(14%)
Film1L_60_50_3_3	60	50	3/3	23(79%)	4(14%)	2(7%)
Film1L_60_50_5_1	60	50	5/1	21(72%)	4(14%)	4(14%)
Film1L_60_50_5_3	60	50	5/3	23(79%)	4(14%)	2(7%)
Film1L_60_60_3_1	60	60	3/1	21(72%)	4(14%)	4(14%)
Film1L_60_60_3_3	60	60	3/3	22(76%)	4(14%)	3(10%)
Film1L_60_60_5_1	60	60	5/1	22(76%)	4(14%)	3(10%)
Film1L_60_60_5_3	60	60	5/3	23(79%)	4(14%)	2(7%)
Film1L_60_70_3_1	60	70	3/1	21(72%)	4(14%)	4(14%)
Film1L_60_70_3_3	60	70	3/3	21(72%)	4(14%)	4(14%)
Film1L_60_70_5_1	60	70	5/1	20 (69%)	4(14%)	5(17%)
Film1L_60_70_5_3	60	70	5/3	20 (69%)	4(14%)	5(17%)

Film 1 – kamera prawa – 29 klatek

Nazwa filmu	Próg koloru czarnego	Próg binaryzacji	Wielkość El. Strukturalnego nr 1 / nr 2	Liczba klatek poprawnych (%)	Liczba klatek niemożliwych (%)	Liczba klatek złych (%)
Film1P_40_50_3_1	40	50	3/1	21(72%)	5(17%)	3(10%)
Film1P_40_50_3_3	40	50	3/3	21(72%)	5(17%)	3(10%)
Film1P_40_50_5_1	40	50	5/1	20 (69%)	5(17%)	4(14%)
Film1P_40_50_5_3	40	50	5/3	20 (69%)	5(17%)	4(14%)
Film1P_40_60_3_1	40	60	3/1	20 (69%)	5(17%)	4(14%)
Film1P_40_60_3_3	40	60	3/3	20 (69%)	5(17%)	4(14%)
Film1P_40_60_5_1	40	60	5/1	20 (69%)	5(17%)	4(14%)
Film1P_40_60_5_3	40	60	5/3	20 (69%)	5(17%)	4(14%)
Film1P_40_70_3_1	40	70	3/1	20 (69%)	5(17%)	4(14%)
Film1P_40_70_3_3	40	70	3/3	20 (69%)	5(17%)	4(14%)
Film1P_40_70_5_1	40	70	5/1	18(62%)	5(17%)	6(20%)
Film1P_40_70_5_3	40	70	5/3	19 (64%)	5(17%)	5(17%)
Film1P_60_50_3_1	60	50	3/1	21(72%)	5(17%)	3(10%)
Film1P_60_50_3_3	60	50	3/3	22(76%)	5(17%)	2(7%)
Film1P_60_50_5_1	60	50	5/1	22(76%)	5(17%)	2(7%)
Film1P_60_50_5_3	60	50	5/3	22(76%)	5(17%)	2(7%)
Film1P_60_60_3_1	60	60	3/1	21(72%)	5(17%)	3(10%)
Film1P_60_60_3_3	60	60	3/3	22(76%)	5(17%)	2(7%)
Film1P_60_60_5_1	60	60	5/1	21(72%)	5(17%)	3(10%)
Film1P_60_60_5_3	60	60	5/3	20 (69%)	5(17%)	4(14%)
Film1P_60_70_3_1	60	70	3/1	21(72%)	5(17%)	3(10%)
Film1P_60_70_3_3	60	70	3/3	21(72%)	5(17%)	3(10%)
Film1P_60_70_5_1	60	70	5/1	21(72%)	5(17%)	3(10%)
Film1P_60_70_5_3	60	70	5/3	21(72%)	5(17%)	3(10%)

Błędne wykrycia prostokąta widoczne na zarejestrowanych filmach są powodowane przez szereg czynników. Najczęściej jest to rozmycie obrazu. W takim przypadku szukany prostokąt nie tylko zmienia kształt na obrazie, ale także jego kolor zostaje wymieszany z jasnym otoczeniem. Algorytm poszukuje narożników prostokąta zakładając, że ten w żaden sposób nie jest obrócony – stąd pojawiające się problemy przy zmianie kształtu poszukiwanego obiektu. Z kolei zmiana koloru na bliższy otoczeniu powoduje mniejszą różnicę po odjęciu otoczenia od obrazu aktualnego – w takim przypadku istnieje duża szansa, że algorytm w trakcie binaryzacji zakwalifikuje piksele należące do prostokąta, jako otoczenie, a nie obiekt.

W filmie zarejestrowanym przez kamerę prawą zachodzi też sytuacja, w której helikopter znajduje się w górnej części obrazu, gdzie jego tłem jest ciemna zasłona. W takim przypadku prostokąt jest niemożliwy do wykrycia. Wynika to z faktu, że ciemna zasłona, podobnie jak prostokąt ma bardzo mało składowej czerwonej. Po odjęciu obrazów otrzymana wartości pikseli prostokąta będą bardzo małe, co spowoduje ich odrzucenie przy binaryzacji.

Na filmach widoczne są też niewielkie przemieszczenia punktów znajdujących przez algorytm mimo braku poruszania się helikoptera. Wynikają one ze zmiennego natężenia światła – zarówno naturalnego, jak i powstającego w wyniku migania diody.

Kolejny analizowany film został nagrany przy tych samych warunkach świetlnych i ustawieniu kamer. Został wybrany z powodu wykonania na nim przez helikopter pełnego obrotu. Kilka klatek filmu zostało pominiętych przy analizie – helikopter nie znajduje się wtedy w polu widzenia kamer.

Film 3 – kamera lewa – 43 klatki (40 z helikopterem)

Nazwa filmu	Próg koloru czarnego	Próg binaryzacji	Wielkość El. Strukturalnego nr 1 / nr 2	Liczba klatek poprawnych (%)	Liczba klatek niemożliwych (%)	Liczba klatek złych (%)
Film3L_40_50_3_1	40	50	3/1	30(75%)	3(8%)	7(17%)
Film3L_40_50_3_3	40	50	3/3	31(77%)	3(8%)	6(15%)
Film3L_40_50_5_1	40	50	5/1	31(77%)	3(8%)	6(15%)
Film3L_40_50_5_3	40	50	5/3	31(77%)	3(8%)	6(15%)
Film3L_40_60_3_1	40	60	3/1	29(72%)	3(8%)	8(20%)
Film3L_40_60_3_3	40	60	3/3	30(75%)	3(8%)	7(17%)
Film3L_40_60_5_1	40	60	5/1	29(72%)	3(8%)	8(20%)
Film3L_40_60_5_3	40	60	5/3	30(75%)	3(8%)	7(17%)
Film3L_40_70_3_1	40	70	3/1	30(75%)	3(8%)	7(17%)
Film3L_40_70_3_3	40	70	3/3	30(75%)	3(8%)	7(17%)
Film3L_40_70_5_1	40	70	5/1	29(72%)	3(8%)	8(20%)
Film3L_40_70_5_3	40	70	5/3	30(75%)	3(8%)	7(17%)
Film3L_60_50_3_1	60	50	3/1	31(77%)	3(8%)	6(15%)
Film3L_60_50_3_3	60	50	3/3	33(82%)	3(8%)	4(10%)
Film3L_60_50_5_1	60	50	5/1	33(82%)	3(8%)	4(10%)
Film3L_60_50_5_3	60	50	5/3	35(87%)	3(8%)	2(5%)
Film3L_60_60_3_1	60	60	3/1	31(77%)	3(8%)	6(15%)
Film3L_60_60_3_3	60	60	3/3	31(77%)	3(8%)	6(15%)
Film3L_60_60_5_1	60	60	5/1	32(79%)	3(8%)	5(13%)
Film3L_60_60_5_3	60	60	5/3	31(77%)	3(8%)	6(15%)
Film3L_60_70_3_1	60	70	3/1	32(79%)	3(8%)	5(13%)
Film3L_60_70_3_3	60	70	3/3	32(79%)	3(8%)	5(13%)
Film3L_60_70_5_1	60	70	5/1	32(79%)	3(8%)	5(13%)
Film3L_60_70_5_3	60	70	5/3	31(77%)	3(8%)	6(15%)

Film 3 – kamera prawa – 43 klatki (40 z helikopterem)

Nazwa filmu	Próg koloru czarnego	Próg binaryzacji	Wielkość El. Strukturalnego nr 1 / nr 2	Liczba klatek poprawnych (%)	Liczba klatek niemożliwych (%)	Liczba klatek złych (%)
Film3P_40_50_3_1	40	50	3/1	21(52%)	3(8%)	16(40%)
Film3P_40_50_3_3	40	50	3/3	23(57%)	3(8%)	14(35%)
Film3P_40_50_5_1	40	50	5/1	24(60%)	3(8%)	13(32%)
Film3P_40_50_5_3	40	50	5/3	24(60%)	3(8%)	13(32%)
Film3P_40_60_3_1	40	60	3/1	24(60%)	3(8%)	13(32%)
Film3P_40_60_3_3	40	60	3/3	25(62%)	3(8%)	12(30%)
Film3P_40_60_5_1	40	60	5/1	25(62%)	3(8%)	12(30%)
Film3P_40_60_5_3	40	60	5/3	26(65%)	3(8%)	11(27%)
Film3P_40_70_3_1	40	70	3/1	27(67%)	3(8%)	10(25%)
Film3P_40_70_3_3	40	70	3/3	27(67%)	3(8%)	10(25%)
Film3P_40_70_5_1	40	70	5/1	28(70%)	3(8%)	9(22%)
Film3P_40_70_5_3	40	70	5/3	27(67%)	3(8%)	10(25%)
Film3P_60_50_3_1	60	50	3/1	24(60%)	3(8%)	13(32%)
Film3P_60_50_3_3	60	50	3/3	24(60%)	3(8%)	13(32%)
Film3P_60_50_5_1	60	50	5/1	25(62%)	3(8%)	12(30%)
Film3P_60_50_5_3	60	50	5/3	24(60%)	3(8%)	13(32%)
Film3P_60_60_3_1	60	60	3/1	27(67%)	3(8%)	10(25%)
Film3P_60_60_3_3	60	60	3/3	26(65%)	3(8%)	11(27%)
Film3P_60_60_5_1	60	60	5/1	27(67%)	3(8%)	10(25%)
Film3P_60_60_5_3	60	60	5/3	27(67%)	3(8%)	10(25%)
Film3P_60_70_3_1	60	70	3/1	27(67%)	3(8%)	10(25%)
Film3P_60_70_3_3	60	70	3/3	28(70%)	3(8%)	9(22%)
Film3P_60_70_5_1	60	70	5/1	28(70%)	3(8%)	9(22%)
Film3P_60_70_5_3	60	70	5/3	28(70%)	3(8%)	9(22%)

Analizując wyniki uzyskane z algorytmu dla filmu nr 3 można zauważyć, że lepsze wyniki uzyskuje się, gdy helikopter zwrócony jest w kierunku kamery prostokątem w kolorze zielonym. Przez dłuższą część filmu helikopter zwrócony jest do kamery niebieską stroną, co spowodowało pogorszeniem wyników w porównaniu do poprzedniego filmu. Najprawdopodobniej wynika to z faktu, że niebieski prostokąt jest zbyt ciemny i niektóre jego piksele są kwalifikowane, jako kolor czarny i pomijane na późniejszym etapie działania algorytmu.

Podsumowując analizę dla wszystkich czterech nagrań najlepsze wyniki otrzymano dla progu koloru czarnego równego 60, progu binaryzacji 50 oraz elementami strukturalnymi dla operacji otwarcia o wielkości 5 i 3. Warto zaznaczyć jednak, że dla drugiego z analizowanych filmów lepsze wyniki uzyskano dla innego zestawu parametrów – odpowiednio 60,70, 3 i 3.

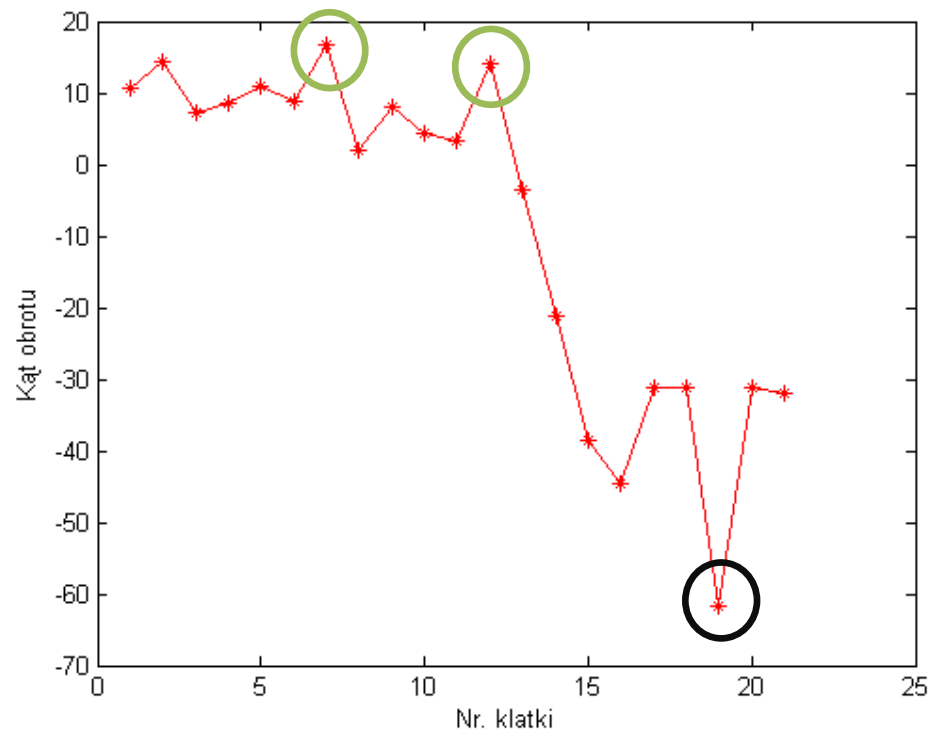
Dla takich dwóch zestawów parametrów postanowiono przeanalizować drugi z algorytmów, czyli pozyskiwanie orientacji helikoptera na podstawie znalezionych prostokątów. Do testów użyto tylko tych klatek, które zostały poprawnie rozpoznane na obrazie oraz ich odpowiednik z drugiej kamery też został rozpoznany poprawnie.

Ponadto algorytm został wzbogacony o dodatkowy parametr. Przy obróbce obrazu często ucinane są piksele tworzące krawędzie prostokąta. Są one najczęściej obcinane nie proporcjonalnie do długości krawędzi prostokąta, a w równej ilości zarówno od wysokości, jak i szerokości figury. Zaburza to wykorzystywany przez algorytm stosunek boków znalezionej figury. Skutkuje to otrzymaniem wartości sinusa szukanego kąta powyżej 1. W celu zniwelowania tegoż efektu otrzymany stosunek boków dzielony jest przez większą od 1 wartość parametru k .

Ocena orientacji nie jest łatwa – w trakcie nagrywania lot nie był w żaden sposób mierzony. Jedynym kryterium jest oszacowanie na podstawie widoku z obu kamer, jaki powinien być kąt obrotu otrzymany z algorytmu.

W przypadku pierwszego z filmów helikopter startuje z kątem obrotu bliskim orientacji zerowej, po czym wykonuje niewielki obrót zgodnie z ruchem wskazówek zegara, który pod koniec jest lekko korygowany przeciwnie do ruchu wskazówek zegara. Algorytm

powinien obliczyć kat początkowy jako bliski zera, a w dalszej części jego wartość powinna maleć i zejść poniżej zera. Pod koniec filmu wartość powinna nieco wzrosnąć.

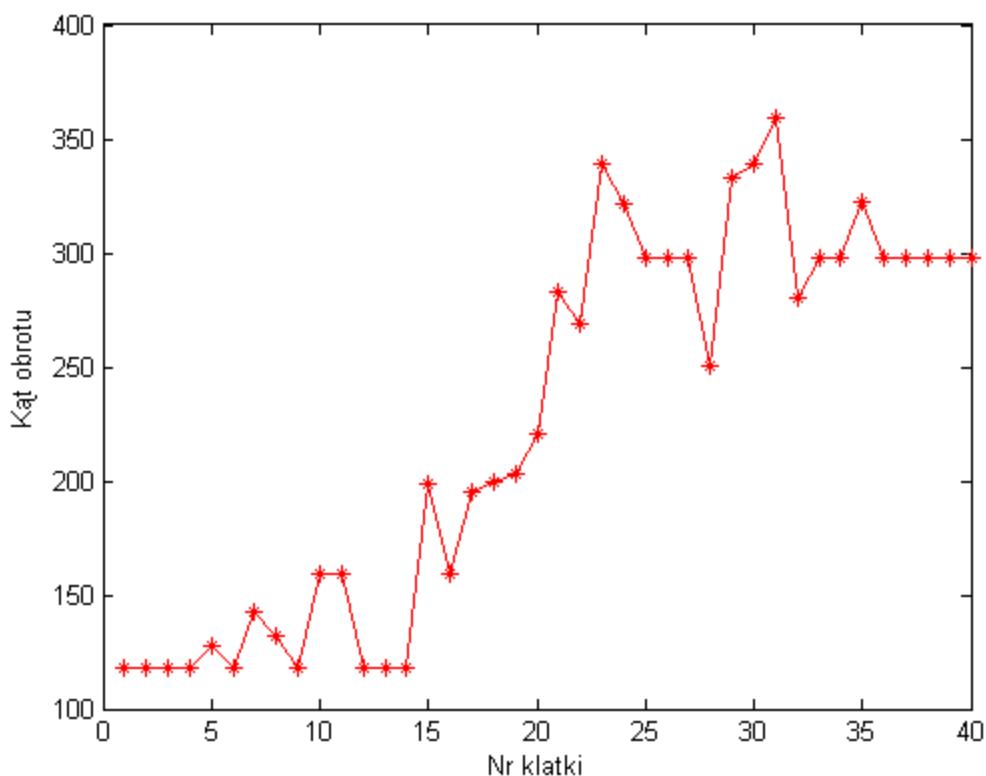


Rysunek 4.11- Orientacja helikoptera na poszczególnych klatkach filmu (Film1, $k = 1.25$)

Na powyższym wykresie widać przebieg kąta obrotu helikoptera obliczonego przez algorytm. W pierwszej fazie występuje brak obrotu, a pewne rozbieżności między wartościami uzyskanymi przez algorytm wynikają z rozmycia, zmian oświetlenia i ostrości kamery. W dalszej części kąt maleje – oznacza to obrót helikoptera w kierunku ruchu wskazówek zegara. W ostatniej fazie wartość kąta obrotu rośnie, co oznacza obrót helikoptera przeciwny do ruchu wskazówek zegara. Opis wywnioskowany z powyższego rysunku pokrywa się z tym, co zaobserwowano na filmach nagranych przez kamery.

W ostatniej fazie lotu występuje duże odstępstwo – oznaczone kolorem czarnym. Jest to niepoprawnie obliczona orientacja – najprawdopodobniej w wyniku błędu przy sprawdzaniu, czy obie kamery są w tej samej ćwiartce układu współrzędnych helikoptera.

Drugi z analizowanych filmów jest trudniejszy do analizy – helikopter wykonuje duży obrót w kierunku przeciwnym do ruchu wskazówek zegara, po którym wykonuje niewielki obrót w kierunku zgodnym z ruchem wskazówek zegara. Początkowy kąt obrotu helikoptera wynosi niespełna 150° .



Rysunek 4.12 - Orientacja helikoptera na poszczególnych klatkach filmu (Film3, $k = 1.25$)

Na powyższym wykresie ujęte zostały wszystkie klatki, na których widoczny był helikopter – również te, na których prostokąt wykryty został niepoprawnie. Początkowy kąt obrotu jest na poziomie około 120° . Następnie helikopter kąt rośnie, co oznacza obrót helikoptera przeciwnie do ruchu wskazówek zegara. W ostatniej fazie lotu widać zmniejszanie się obliczonej przez algorytm wartości kąta, co oznacza obrót helikoptera w przeciwnym kierunku.

Helikopter na filmie jest często rozmyty, a prostokąt na niektórych klatkach jest prawie niewidoczny – mimo tego algorytm poprawnie obliczył orientację śmigłowca. Nie udało się jednak uniknąć błędów – kilka punktów wyraźnie odbiega od rzeczywistej orientacji. Warto także zauważyć, że w przypadku braku ruchu można zaobserwować kilka takich samych wartości kąta – oznacza to dobra powtarzalność systemu.

5. Moduł komunikacji

5.1 Wstęp

Podsumowując poprzedni rozdział udało się stworzyć system pozwalający na odczyt wszystkich informacji opisujących stan helikoptera w danej chwili czasowej. Z punktu widzenia automatyki analizowany obiekt, jakim jest śmigłowiec posiada już wyjścia. Jednak wciąż nie można nim w żaden sposób sterować – potrzebne jest wejście dla sygnałów sterujących.

Jak wspomniano wcześniej niewielkie rozmiary i waga użytej jednostki bardzo zawężają możliwości instalowania dodatkowych elementów. Producent w celu komunikacji zamontował w helikopterze odbiornik podczerwieni oraz diody IR w pilocie. Do komunikacji bez udziału człowieka można wykorzystać odbiornik, ale pilot nie jest w stanie sam wysyłać poleceń do śmigłowca. W ramach pracy zdecydowano się więc na stworzenie nadajnika podłączonego do komputera emitującego taki sam sygnał, jaki wysyła pilot.

Nadajnik ten powinien w prosty sposób komunikować się z komputerem. Dla systemu windows istnieje wiele różnych bibliotek pozwalających na obsługę urządzeń podczerwonych. Jedną z prostszych i bardziej intuicyjnych jest wykorzystana w pracy biblioteka WinLIRC. Dzięki niej możliwa jest obsługa nadajników i odbiorników podczerwieni komunikujących się z komputerem przez łącze szeregowo RS-232.

Po skonstruowaniu nadajnika należy także zbadać, w jaki sposób sekwencja załączania i gaszenia diody jest analizowana przez odbiornik helikoptera w celu pozyskania informacji. Niezbędna jest identyfikacja protokołu komunikacji. W tym celu zdecydowano się przyrzeć sygnałowi wysyланemu przez pilota za pomocą prostego odbiornika podczerwieni zbudowanego na potrzeby pracy. Odbiornik ten również współpracuje z biblioteką WinLIRC.

5.2 Odbiornik IR

Pierwszym z konstruowanych układów ma za zadanie odebrać i przesłać do komputera sygnał podczerwony. Kluczowym elementem takiego schematu jest odbiornik CIR, czyli Consumer Infrared. Określeniem tym nazywa się szeroką gamę urządzeń używających fal podczerwonych do komunikacji bezprzewodowej. Stosuje się je przede wszystkim w pilotach telewizyjnych, czy bezprzewodowych klawiaturach i myszach.

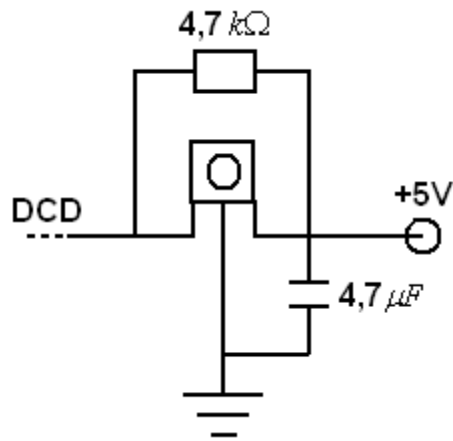
Odbiornik ten nie składa się tylko z foto-tranzystora reagującego na pasmo podczerwieni. Zawiera także obwód demodulacji oscylujący wokół określonej częstotliwości nośnej (najczęściej 38kHz) oraz zależny od niego układ zatrasku wyjścia. Dodatkowe obwody mają na celu blokowanie światła pochodzącego z otoczenia oraz szumów w celu usprawnienia odczytu danych. W celu przesłania informacji dioda nadawcza – czy to w pilocie do telewizora, czy do helikoptera – zapala się i gaśnie w odpowiednich chwilach czasowych według odpowiedniego protokołu. Jednak po zapaleniu dioda nie świeci się, a miga z częstotliwością nośną. Gdyby świecenie było ciągle zostałoby uznane przez odbiornik za światło z otoczenia, a co za tym idzie zignorowane. Ponieważ jednak światło jest zmodulowane z odpowiednią częstotliwością układ demodulacji odbiornika zareaguje załączeniem układu zatrasku, co spowoduje włączenie się pinu wyjściowego. Po wyłączeniu nadawania sygnału układ demodulacji zwolni układ zatrasku, a on wyłączy pin wyjściowy.

Zatrask występujący w odbiorniku nie działa jednak natychmiastowo na zmiany sygnału. Większość dostępnych modeli nie reaguje na sygnał zanim nie powtórzy się on co najmniej 10 razy. Po krótkim rachunku można stwierdzić, że przy użytych 38 kHz jako częstotliwość nośna najkrótszy rozpoznany sygnał może mieć $260\mu\text{s}$. Taka sama sytuacja występuje po zakończeniu migania diody – jeszcze przez około 10 cykli zatrask trzyma wyjście układu w stanie wysokim. Sytuacja ta stworzona jest celowo i ma na celu uodpornienie odbiornika na zakłócenia. Z drugiej strony układy wymagające szybkiej reakcji albo takie, w których występują bardzo krótkie sygnały mogą napotkać problemy przy użytkowaniu takiego odbiornika

Z uwagi na to, że w dany model odbiornika CIR posiada modulacje z konkretną częstotliwością ważne jest, aby była ona dopasowana do nadajnika. Nie oznacza to jednak, że niewielkie różnice w częstotliwościach uniemożliwią działanie układu.

Odbiornik z jest w stanie odczytać informacje nadawane z częstotliwościami o 10-20% większymi bądź mniejszymi od jego częstości demodulacji. Dzieje się to jednak kosztem zasięgu poprawnego rozpoznawania informacji oraz wymaga dłuższych sygnałów minimalnych.

W pracy zdecydowano się na niezbyt skomplikowany układ przedstawiony poniżej:



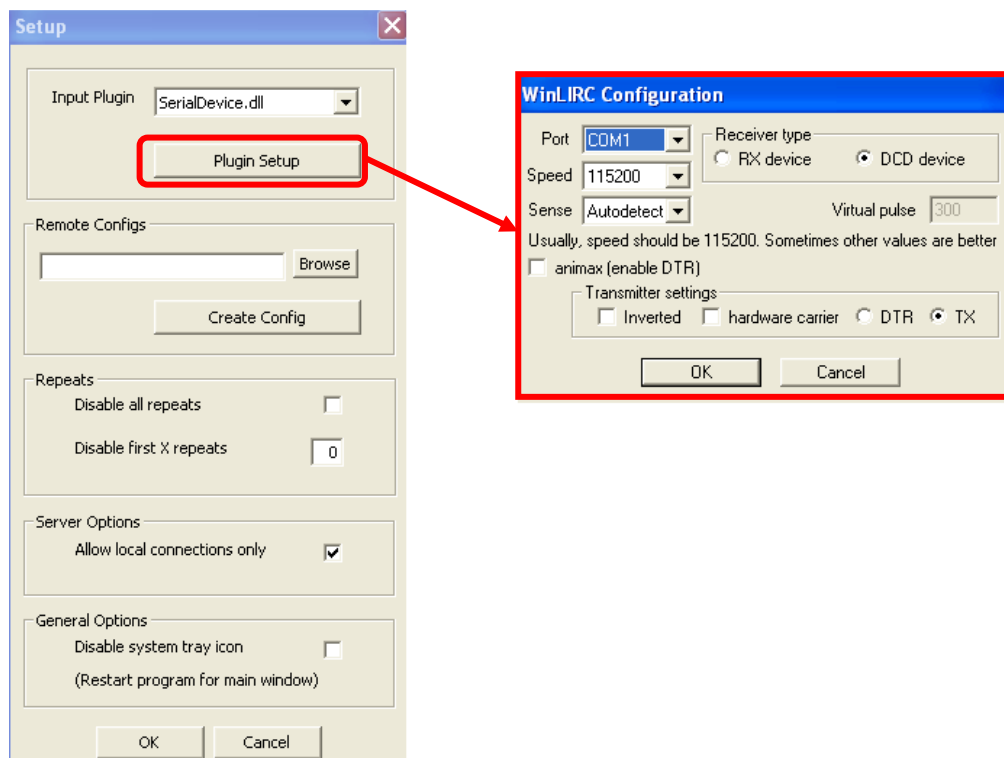
Rysunek 5.1 - Schemat elektryczny odbiornika

W skład stworzonego układu wchodzi opornik o wartości rezystancji równej $4,7\text{k}\Omega$, kondensator o pojemności $4,7\mu\text{F}$ i odbiornik podczerwieni TSOP1738 firmy Vishay. Model ten spełnia wszystkie wymienione wymagania – częstotliwość modulacji wynosi 38 kHz, a foto-tranzystor reaguje najlepiej na fale o długości 950nm. Ponadto TSOP1738 jest wymieniony na liście odbiorników przetestowanych z biblioteką WinLIRC.

Wśród sygnałów zaznaczonych na schemacie można wyróżnić masę oraz zasilanie i sygnał DCD. Uzyskanie stałego zasilania +5V używając łącza RS-232 nie jest proste – wymagany jest regulator napięcia, który mocno komplikuje cały układ. W celu pozyskania takiego napięcia zdecydowano się na użycie złącza USB, które dysponuje właśnie takim potencjałem. Masę ze złącza USB należy spiąć wraz z masą łącza RS-232, czyli pinem GND oraz z masą zaznaczoną na schemacie 4.1. Ostatnim z zaznaczonych sygnałów jest DCD, czyli Data Carrier Detected. Jest to jeden z sygnałów łącza RS-232 wykorzystywany do komunikacji przez bibliotekę WinLIRC.

Zasada działania układu jest bardzo prosta. W przypadku, gdy do odbiornika dochodzi sygnał podczerwony, jego wyjście ustawione jest na GND, co powoduje przepływ prądu przez rezystor. Dzięki temu pin DCD łącza szeregowego zostaje ściągnięty do 0V. W przypadku, gdy odbiornik nie dostaje żadnego sygnału, na jego wyjściu pojawia się stan wysoki – napięcie bliskie napięciu zasilania. W ten sposób na pinie DCD też pojawia się napięcie bliskie 5V. Jak widać w układzie fakt pojawienia się sygnału podczerwonego, bądź jego braku skutkuje zmianą napięcia na łączy szeregowym w zakresie 0V/5V. Jak wiadomo wartości napięć występujące w RS-232 są symetryczne względem zera. Jednak logika, jaką posługuje się komputer przy czytaniu łącza szeregowego pozwala na poprawne odczytanie sygnałów. Dla napięcia 5V nie występuje problem – jest ono rozpoznawane jako logiczne „0” przez port szeregowy, ponieważ znajduje się w domyślnym zakresie 3÷12V. Wartość 0V nie należy do domyślnego przedziału -12÷-3V, ale zostaje rozpoznane jako logiczne „1”, ponieważ nie należy do przedziału zarezerwowanego dla wartości „0”.

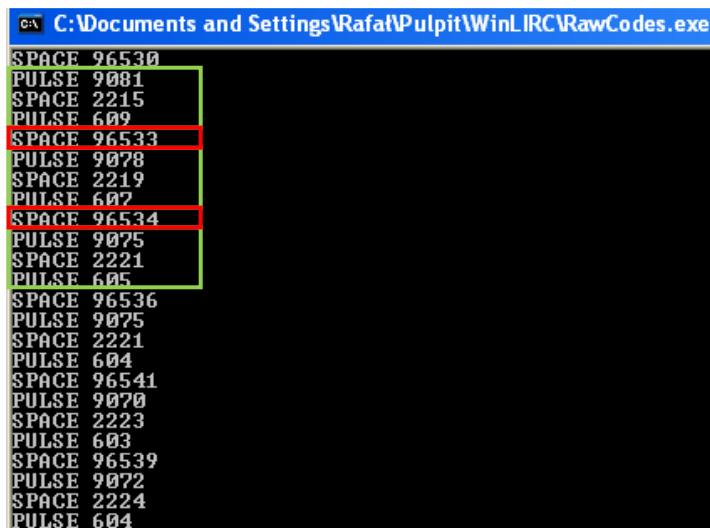
Po skonstruowaniu układu należało go przetestować. W tym celu wykorzystano program wchodzący w skład biblioteki WinLIRC. Po jego uruchomieniu należało odpowiednio ustawić wszystkie parametry. Do testu samego odbiornika wystarczy wybrać łącze szeregowe, a w oknie jego konfiguracji ustawić port, do którego podłączone zostało urządzenie i jego typ, czyli w przypadku pracy „DCD device”.



Rysunek 5.2 - Okno konfiguracji programu i okno konfiguracji urządzenia

Następnie należało zamknąć cały program, a następnie uruchomić jeden z mniejszych programów dostarczonych z biblioteką, czyli RawCodes. Program ten pozwala na bezpośrednie obserwowanie sygnału, jaki przesyła odbiornik bez jakiegokolwiek analizy. Do testu użyto pilota telewizyjnego – naciśnięcie i przytrzymanie przycisku powinno zostać odczytane jako kilkukrotnie powtarzana sekwencja.

Uruchomiono program, następnie skierowano w stronę odbiornika pilot i przytrzymano przycisk Prog+ na pilocie. Otrzymano następujący ekran:



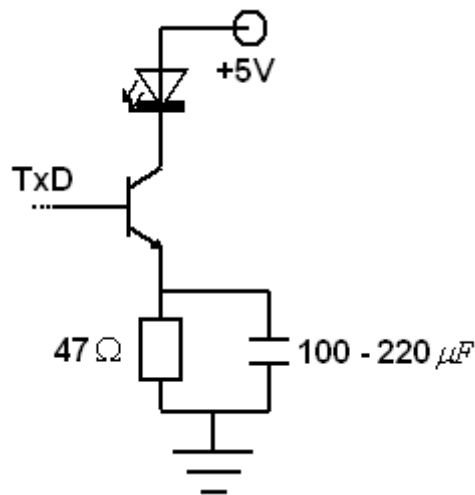
```
C:\Documents and Settings\Rafał\Pulpit\WinLIRC\RawCodes.exe
SPACE 96530
PULSE 9081
SPACE 2215
PULSE 609
SPACE 96533
PULSE 9078
SPACE 2219
PULSE 607
SPACE 96534
PULSE 9075
SPACE 2221
PULSE 605
SPACE 96536
PULSE 9075
SPACE 2221
PULSE 604
SPACE 96541
PULSE 9070
SPACE 2223
PULSE 603
SPACE 96539
PULSE 9072
SPACE 2224
PULSE 604
```

Rysunek 5.3 - Ekran aplikacji RawCodes

Program wypisuje na ekranie odczytaną akcję wraz z czasem jej trwania podanym w mikrosekundach. Przez puls program rozumie fakt, że odbiornik widzi nadawanie sygnału. Przerwa oznacza brak odczytanego sygnału. Analizując uzyskane wyniki można zauważyć powtarzającą się sekwencje sygnałów. Pierwsze trzy sygnały oznaczone na zielono to nagłówek w postaci pulsu o długości 9070 i przerwy o długości 2220 oraz komenda w postaci pulsu o długości 600. Następnie występuje przerwa oznaczona na czerwono, która wynika z mechaniki działania pilota jako najkrótszy czas między kolejnymi przesłaniami komendy. Nie wglębiano się bardziej w zasadę działania pilota, gdyż nie jest to przedmiotem pracy – powtarzający się sygnał przy naciśnięciu tego samego przycisku jednoznacznie oznacza, że odbiornik działa poprawnie.

5.3 Nadajnik IR

Drugi z projektowanych układów ma za zadanie za pomocą diody IR przesyłać sygnał do helikoptera. W literaturze spotyka się kilka rozwiązań, lecz w pracy zdecydowano się na schemat o niskim poziomie skomplikowania. Układ jest zbudowany tylko z niezbędnych elementów.



Rysunek 5.4 - Schemat elektryczny nadajnika

W skład stworzonego układu wchodzi tranzystor bipolarny typu npn model BC547, rezystor 47Ω mający na celu ustalenie odpowiedniego prądu płynącego w układzie oraz kondensator niwelujący wpływ szumów. Kondensator użyty w pracy ma pojemność 220μF. Przy budowie wykorzystano diodę podczerwieni LD271 firmy OSRAM. Użyty model najsilniej emituje fale o długości 950nm. Jest to najpopularniejsza długość fali wykorzystywana przy sterowaniu podczerwienią. Model ten dysponuje też szerokim kątem emisji, co w przypadku poruszającego się odbiornika jest kluczowe.

Nadajniki podczerwieni LED najczęściej nie mają problemów z częstotliwością sygnału, które występują w przypadku omówionych wcześniej odbiorników. Wynika to z faktu, że ich budowa jest bardzo prosta, a co za tym idzie posiadają bardzo krótki czas odpowiedzi. Ponadto nie muszą one dekodować danych ani bronić się przed zakłóceniami. Jeżeli tylko urządzenie sterujące nadajnikiem – w przypadku pracy jest to komputer – posiada odpowiedni zegar oraz wszystkie niezbędne dane praca układu nadawczego nie stwarza żadnych problemów.

Na schemacie wyróżniono sygnały, dzięki którym możliwa jest realizacja nadajnika. Pierwszym z nich jest masa. Do masy tej w przypadku użycia 9-pinowego łącza RS-232 należy podpiąć pin o numerze 5 – w ten sposób stworzono jedną, wspólną masę. Kolejnym sygnałem jest zasilanie, czyli +5V widoczne u góry schematu. Uzyskanie stałego napięcia o takiej wartości z łącza szeregowego nie jest proste, więc zdecydowano się na poprowadzenie napięcia zasilania za pomocą kabla USB. W takim przypadku masę występującą w kablu USB należy również połączyć z pozostałymi masami.

Ostatnim sygnałem wymagającym omówienia jest doprowadzony do bazy tranzystora sygnał TxD, który występuje na trzecim pinie zastosowanego łącza szeregowego. Sygnał ten odpowiada za transmisję danych. Standard WinLIRC pozwala na konstrukcję nadajników z użyciem sygnałów TxD, bądź DTR. Rekomendowane jest jednak użycie sygnału wykorzystanego w pracy. W przypadku wysłania przez komputer logicznego „0” do schematu trafia dodatkowo napięcie. Oznacza to, że przez diodę prąd płynie w kierunku zaporowym, co powoduje brak emisji fali. W przypadku logicznej „1” sygnał TxD przyjmuje wartość ujemną co powoduje przepływ prądu w kierunku przewodzenia, co skutkuje emisją fal podczerwonych.

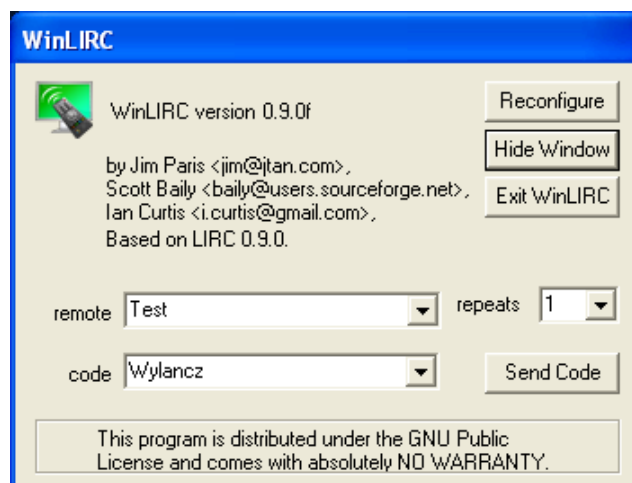
Po skonstruowaniu układu według omówionego schematu należało go przetestować. Napotkano jednak drobne utrudnienie – człowiek nie jest w stanie dostrzec fal podczerwonych, czyli niemożliwe jest stwierdzenie czy dioda świeci, czy też nie. W celu obserwacji diody podczerwonej użyto jednej z kamer wykorzystywanych w pracy. Kamery nie posiadają filtra podczerwieni, co oznacza, że fale o częstotliwościach niższych niż kolor czerwony są rejestrowane na matrycy, a co za tym idzie są widoczne na obrazie. Jednak w ten sposób można stwierdzić jedynie, czy dioda świeci, czy nie, ale nic nie można powiedzieć o poprawności przesłanego sygnału.

Dzięki przetestowanemu już odbiornikowi można stworzyć układ pozwalający na odczyt i przesyłanie sygnału podczerwonego. Wiedząc, że część odbiorcza działa poprawnie można w prosty sposób sprawdzić poprawność nadawanego sygnału za pomocą programu wchodzącego w skład biblioteki WinLIRC. Testem było odczytanie i zapamiętanie sygnału wysyłanego z pilota telewizyjnego, a następnie przesłanie

takiego samego sygnału z diody nadawczej skierowanej w stronę telewizora. Łatwo można zaobserwować wyniki – podczas testu wykorzystano przycisk wyłączający telewizor.

Po ustawieniu opcji programu jak w przypadku sprawdzania nadajnik należało stworzyć plik konfiguracyjny dla zastosowanego pilota. Po wybraniu tej opcji pojawia się okno z wypisanymi poleceniami dla użytkownika. Należy wprowadzić nazwę dla pilota oraz nacisnąć kilkakrotnie przycisk wraz z nazwaniem jego funkcji. Program po wysłaniu sygnałów dla dwóch przycisków jest w stanie odczytać nagłówek i sygnał właściwy użytego modelu.

Następnie można przejść do okna głównego pozwalającego na wysłanie wybranego sygnału do telewizora. Po wybraniu nazwy pilota wprowadzonej przy konfiguracji należy wybrać komendę, którą nadajnik ma przesłać oraz ilość powtórzeń sygnału. Podczas testu wybrano opcje widoczne na rysunku poniżej.



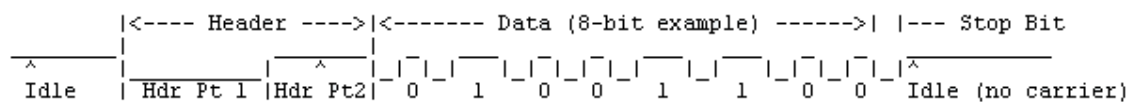
Rysunek 5.5 - Okno główne pozwalające na wysłanie sygnału

Po wysłaniu sygnału do telewizora nastąpiło jego wyłączenie. Warto zauważyć, że wystarczy już jeden sygnał, aby odbiornik w telewizorze odczytał i wykonał komendę. Pomyślne odczytanie przesłanego rozkazu jednoznacznie oznacza, że zarówno odbiornik, jak i nadajnik stworzonego układu działają poprawnie.

5.4 Analiza protokołu komunikacji

Kolejnym krokiem było określenie zasad, na jakich porozumiewa się nadajnik z helikopterem użytym w pracy. Pierwszym pomysłem było pozyskanie informacji od producenta. Mimo wielu prób kontaktu nie udało się uzyskać jakiegokolwiek odpowiedzi od firmy SYMA. Postanowiono więc dokonać interpretacji sygnału za pomocą skonstruowanego odbiornika i oprogramowania dostępnego wraz z użytą biblioteką obsługi podczerwieni.

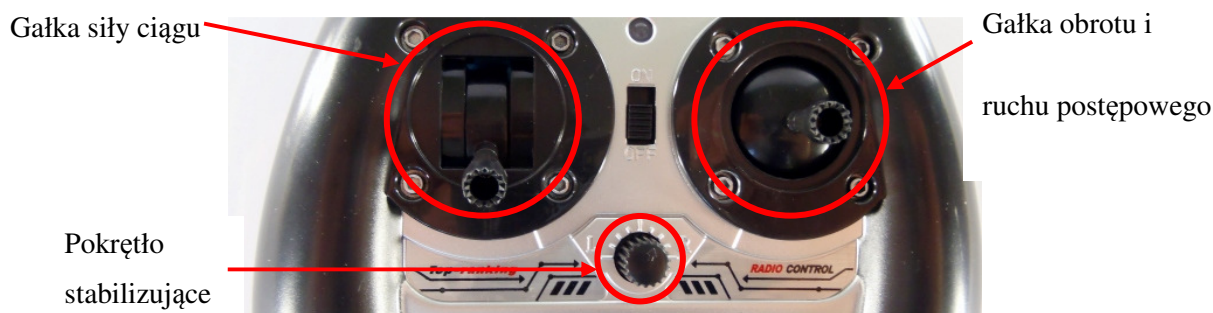
Na podstawie literatury oraz wielu stron internetowych poświęconych zagadnieniu sterowania podczerwienią stwierdzono, że sygnał przesyłany do helikoptera można podzielić na nagłówek w postaci dłuższych sygnałów informujących odbiornik o początku nadawania informacji oraz na odpowiednio kodowane bity kolejnych informacji. W niektórych protokołach na końcu występują odpowiednie bity stopu, ale jest to rozwiązanie rzadko spotykane w przypadku podczerwieni. Poszczególne bity wchodzące w skład przesyłanego sygnału mają wartość 0 albo 1. W protokołach komunikacji podczerwieni wartości te nie są sygnalizowane po prostu przez świecenie diody bądź jego brak. Dla obu wartości występuje na początku miganie diody o stałej długości rozpoznawane przez odbiornik jako sygnał wysoki, a następnie następuje przerwa, której długość decyduje o tym, czy bit zostanie rozpoznany jako 0 czy 1.



Rysunek 5.6 - Prezentacja graficzna przykładowego protokołu komunikacji

Źródło: www.picaxeforum.co.uk

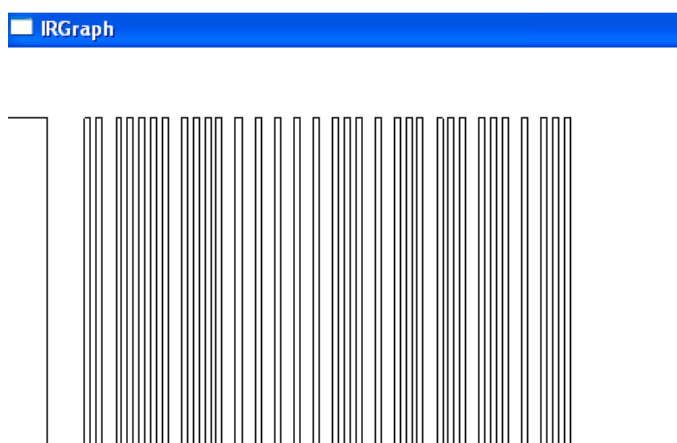
Jak wspomniano na początku pracy producent podaje, że helikopter jest sterowany za pomocą trzech i pół sygnału. Jest to pewne niesprecyzowanie, ponieważ nie da się przesłać połowy sygnału. Producent ma tu na myśli fakt, że użytkownik w trakcie obsługi śmigłowca steruje trzema sygnałami za pomocą gałek w pilocie, a dodatkowa, czwarta informacja jest pomocnicza, pozyskiwana z pokrętła stabilizującego. Według instrukcji przed użytkowaniem helikoptera należy podlecieć do góry na niewielką wysokość i ewentualny obrót jednostki wokół osi pionowej zniwelować właśnie za pomocą tego pokrętła.



Rysunek 5.7 - Pilot do helikoptera SYMA s107

Źródło: syma.com

Dla pracy nie ma to jednak znaczenia – wysyłając sygnał do helikoptera należy przesłać wszystkie cztery wartości. W większości protokołów komunikacji każda informacja kodowana jest z użyciem 8 bitów, co w przypadku użytego śmigłowca daje 32 bity przesłane za pomocą podczerwieni. W literaturze można spotkać protokół komunikacji „2x20”, który charakteryzuje się właśnie takimi parametrami. Na potwierdzenie powyższych rozważań przedstawiono poniżej przykładowy sygnał nadawany z pilota.



Rysunek 5.8 - Przykładowy sygnał z pilota (Dla zwiększenie przejrzystości zmieniona kolor tła na biały)

Na powyższym sygnale łatwo zaobserwować nagłówek, 32 całkowite sygnały, które można zinterpretować jako 0 i 1 oraz krótki puls, po którym następuje długa przerwa – jest to najprawdopodobniej bit stopu, jednak jego zachowanie należy poddać analizie.

Przy analizie protokołu wykorzystano program RawCodes oraz IRGraph wchodzące w skład używanej biblioteki. Zachodzą jednak pewne niedogodności – programów tych nie da się uruchomić jednocześnie, co uniemożliwia graficzne przedstawienie sygnału oraz odczyt konkretnych wartości. Ponadto zauważono, że pilot do wysłania jakiegokolwiek sygnału potrzebuje chociaż niewielkiego wychylenia gałki powodującej ruch pionowy. Ostatnim z napotkanych utrudnień był fakt, że przy odpuszczeniu gałki odpowiedzialnej za siłę ciągu wysyłany jest sygnał w postaci nagłówka i wartości domyślnych z drugiej gałki – niezależnie od jej faktycznego wychylenia.

Pierwszym z analizowanych elementów była długość sygnałów odpowiadających nagłówkowi oraz odpowiednim bitom w sygnale. Zaobserwowano je za pomocą aplikacji RawCodes wysyłając kilkakrotnie sygnał za pomocą pilota do helikoptera. Podczas wysyłania wychylano gałki w celu uzyskania różnych sygnałów z dużą ilością zer i jedynek. Interpretując otrzymane wyniki stwierdzono, że w nagłówku sygnału występuje puls o czasie trwania nieco powyżej 2000 μ s, po którym występuje przerwa o długości niespełna 2000 μ s. W dalszej części sygnału występuje mniejsza powtarzalność – wynika ona z bardzo krótkiego czasu trwania sygnałów. Pulsy występujące w kodowaniu bitów mają długości w zakresie 250-450 μ s. Przerwy w przypadku logicznego 0 są w takim samym zakresie, a w przypadku występowania logicznej 1 długości przerw są w zakresie 550-700 μ s. Końcowy puls występujący w sygnale ma wartości takie jak te występujący w przypadku kodowaniu bitów, a występująca po nim długa przerwa trwa 90,5-90,7 ms. Oznacza to najprawdopodobniej, że bitem stopu jest wartość 1, po której występuje przerwa o długości 90 ms.

Kolejnym analizowanym elementem protokołu była kolejność wysyłania informacji. Zbadano ją za pomocą programu IRGraph. Poruszając obiema gałkami oraz pokrętle stabilizacyjnym można było zaobserwować zmiany w sygnale w konkretnym miejscu. W efekcie stwierdzono, że jako pierwsza przesyłana jest informacja o poprzecznym wychyleniu prawej gałki pilota odpowiedzialna za obrót helikoptera wokół osi pionowej. Następną przesyłaną informacją jest wychylenie pionowej tego samego manipulatora odpowiedzialne za ruch postępowy. Trzecią informacją jest stan lewego drążka sterującego siłą ciągu. Za ostatnią część sygnału odpowiada obrót pokrętła stabilizującego.

Istotną sprawą są przedziały wartości, jakie przyjmują odpowiednie sygnały oraz jakim sytuacją rzeczywistym one odpowiadają. Analizując pierwszą składową sygnału, jaką jest obrót helikoptera, zaobserwowano sygnał z pilota przy wychyleniu gałki w lewo, w prawo oraz na środku – w jej stanie wolnym.



Rysunek 5.9 - Graficzna ilustracja sygnału ze zmiennym wychyleniem poziomym gałki prawej

Łatwo zauważyć, że wartość znajduje się w przedziale 0-127 z punktem zerowym w wartości 64 – niewielkie rozbieżności między zaproponowanymi wartościami, a tymi widocznymi powyżej wynikają z niedokładności mechanicznej urządzenia nadawczego – jest to klasyczny problem przy zamianie analogowego sygnału, jakim jest wychylenie na wartość cyfrową.

Można także zauważyć, że występuje w badanym protokole klasyczne kodowanie liczby binarnej. Założono, że występuje ono także dla pozostałych informacji zawartych w przesyłanym sygnale. Oznacza to, że do poznania zakresu i przebiegu wartości wystarczy tylko dwie wartości – przy pełnym wychyleniu w jedną i w drugą stronę.

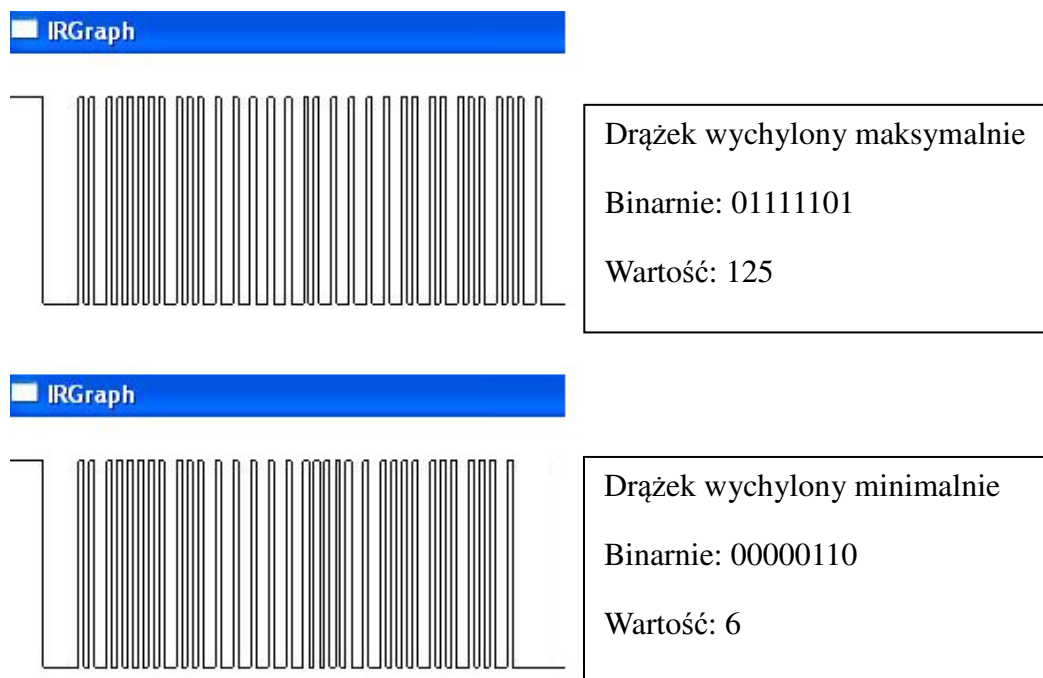
Kolejnym analizowanym sygnałem było wychylenie lewej gałki w pionie odpowiadające za ruch postępowy śmigłowca.



Rysunek 5.10 - Graficzna ilustracja sygnału ze zmiennym wychyleniem pionowym gałki prawej

Podobnie jak w poprzednim przypadku przesyłane wartości są w zakresie 0-127. Ponadto założono, że brak wychylenia gałki, czyli stan wolny jest reprezentowany przez wartość w środku zakresu, czyli 64. Można zauważyć, że w przypadku tej informacji wartości są mało intuicyjne – zwiększenie wartości spowoduje bowiem szybszy ruch helikoptera do tyłu zamiast do przodu.

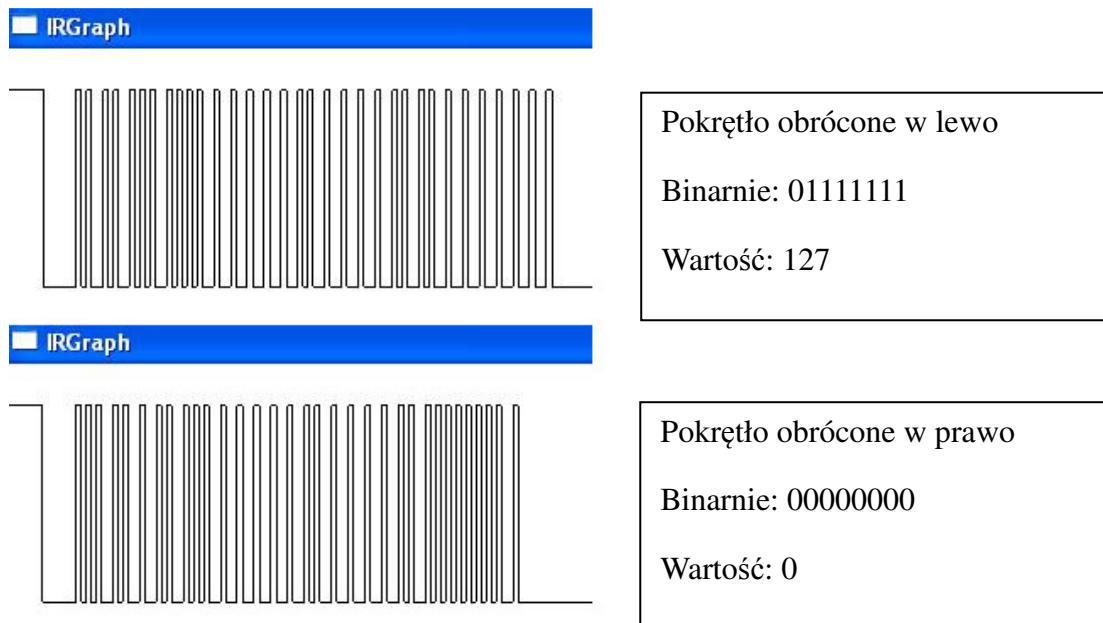
Trzecią informacją odpowiada za siłę ciągu helikoptera. W tym przypadku nie można było zobaczyć sygnału dla zerowego wychylenia drążka – jak wspomiano sygnał wtedy może być błędny. Dokonano więc analizy obserwując sygnał przy maksymalnym oraz bardzo małym wychyleniu.



Rysunek 5.11 - Graficzna ilustracja sygnału ze zmiennym wychyleniem lewego drążka

W przypadku tego sygnału zakres wartości również wynosi 0-127. Nie ma tu jednak stanu wolnego drążka – jego zwolnienie przez użytkownika powoduje powrót do wartości 0. Zachowanie helikoptera jest intuicyjne – im większa wartość sygnału sterującego tym większa siła ciągu wynikająca z obrotu śmigieł.

Ostatnią informacją zawartą w sygnale jest stan pokrętła. Jego wartość będzie rzadko zmieniana, jednak warto poznać ewentualny zakres zmian. W praktyce wartość sygnału stabilizującego znajduje się wokół środka zakresu.



Rysunek 5.12 - Graficzna ilustracja sygnału ze zmiennym obrotem pokrętkła stabilizującego

Zauważono, że wartości są w zakresie 0-127, co oznacza, że wartości używane w pracy będą oscylować wokół środka tegoż zakresu, czyli wartości 64.

Podsumowując dokonano analizy sygnału przesyłanego z pilota do helikoptera. Sygnał ten rozpoczyna się nagłówkiem stworzonym z pulsu trwającego 2ms, po którym występuje równie długa przerwa. Kolejnymi elementami są cztery odpowiednio zakodowane informacje – pierwsze dwie opisują prawą gałkę kontrolera, kolejna wychylenie lewego drążka, a ostatnia ustawienie pokrętkła stabilizującego. Informacje składają się z bitów, które w sygnale są reprezentowane jako krótki puls, po którym następuje przerwa, której długość rozstrzyga czy dany bit ma wartość 0 czy 1. Każda z informacji kodowana jest na 8 bitach, a jej wartość waha się w przedziale od 0 do 127. Po przesłaniu wszystkich 32 bitów następuje dodatkowy bit stopu o wartości 1, po którym występuje przerwa o długości 90 ms.

5.5 Plik konfiguracyjny

Biblioteka WinLIRC działa na zasadzie komend. Przy nadawaniu sygnału wybierana jest jedna z wcześniej stworzonych komend, pod którą zapisany jest dokładny przebieg wysyłanego sygnału. Oznacza to, że na potrzeby pracy należało stworzyć odpowiedni plik konfiguracyjny zawierający wszystkie możliwe komendy przesyłane do helikoptera. W skład biblioteki wchodzi program pozwalający na tworzenie pliku konfiguracyjnego poprzez przesłanie do odbiornika poszczególnych sygnałów, jednak dla pilota posiadającego gałki analogowe trudno jest uzyskać stały sygnał o wybranych współczynnikach.

Liczba komend jest duża, ale skończona. Każdy z czterech sygnałów może przyjmować wartości od 0 do 127, co oznacza, że istnieje 128 różnych wartości dla każdego sygnału. Łatwo policzyć, że do pokrycia wszystkich możliwych sytuacji należy stworzyć 128^4 , czyli ponad 250 milionów komend. Nawet przy ograniczeniu wartości odpowiadającej za pokrętko stabilizujące ręczne wpisywanie takiej ilości komend trwałoby bardzo długo, dlatego stworzono prosty program, który sam wypisuje wszystkie komendy.

Stworzony plik konfiguracyjny wygląda następująco:

```
begin remote
  name Helicopter
  bits 32
  flags SPACE_ENC
  eps 25
  aeps 100

  header 2000 2000
  one 300 600
  zero 300 300
  ptrail 300
  pre_data_bits 0
  pre_data 0x0
  gap 90700
  toggle_bit 0

  begin codes
    0/0/0/0 0x0000000000000000
    0/0/0/1 0x0000000000000001
    ...
    127/127/127/127 0x000000007F7F7F7F
  end codes
end remote
```

Rysunek 5.13 - Plik konfiguracyjny (Pozostałe komendy zastąpiono trójkropkiem)

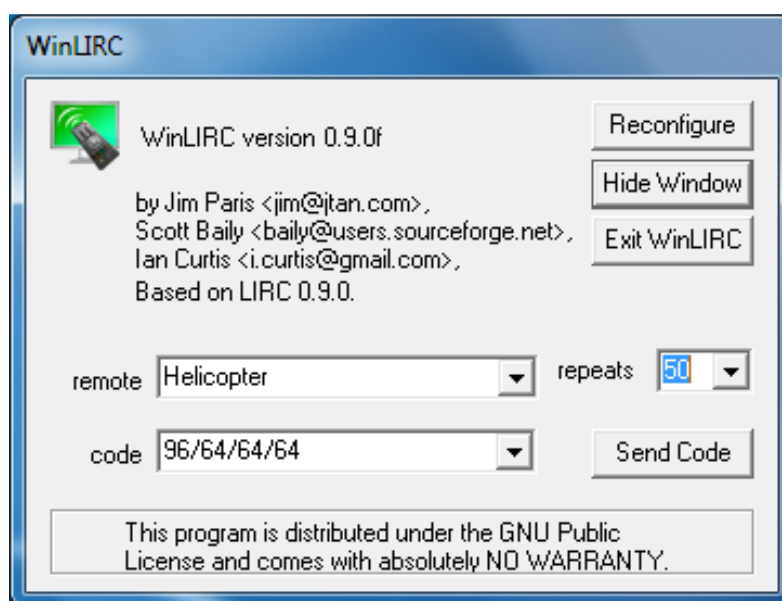
Informacje wchodzące w skład pliku konfiguracyjnego:

- name – nazwa pilota
- bits – liczba bitów danych
- eps – dokładność wyrażana w procentach
- aeaps – dokładność wyrażona w mikrosekundach
- header – nagłówek. Pierwsza wartość to długość trwania pulsu, a druga to długość przerwy po nim występującej
- one – kodowanie logicznej „1”. Pierwsza wartość to długość trwania pulsu, a druga to długość przerwy po nim występującej
- zero – kodowanie logicznego „0”. Pierwsza wartość to długość trwania pulsu, a druga to długość przerwy po nim występującej
- ptrail – długość pulsu następującego po wszystkich danych
- pre_data_bits – liczba bitów występujących przed danymi
- pre_data – wartość bitów występujących przed danymi
- gap – długość przerwy po całym sygnale
- Między begin codes, a end codes wypisane są wszystkie komendy. Po lewej stronie występuje nazwa komendy, a po prawej wartość, jaka ma być w jej ramach wysłana. Wartość wyrażona jest w systemie szesnastkowym

Tak stworzony plik może zostać wykorzystany do przesyłania sygnału bezpośrednio z programu głównego WinLIRC lub z linii komend wykorzystując program Transmit. Jako argumenty do programu Transmit należy podać nazwę pilota, komendy oraz ilość powtórzeń sygnału. Program ten wchodzi w skład wykorzystanej biblioteki.

5.6 Próba nawiązania komunikacji

Kolejnym krokiem po stworzeniu pliku konfiguracyjnego było przesłanie sygnału do helikoptera i zaobserwowanie jego reakcji. Nie wystarczy przesłać jednak pojedynczego sygnału, ale kilkakrotnie powtórzyć ten sam sygnał. Średni czas trwania całego sygnału to około 120 ms. Helikopter na początku komunikacji ma pewne opóźnienie, więc należało wysłać taką liczbę sygnałów, aby czas wysyłania był dużo większy od tegoż opóźnienia. Zdecydowano się przesłać 50 sygnałów, co będzie trwać 6 sekund. W tym celu uruchomiono program główny biblioteki WinLIRC i ustawiono plik konfiguracyjny, wybrano komendę oraz liczbę powtórzeń.



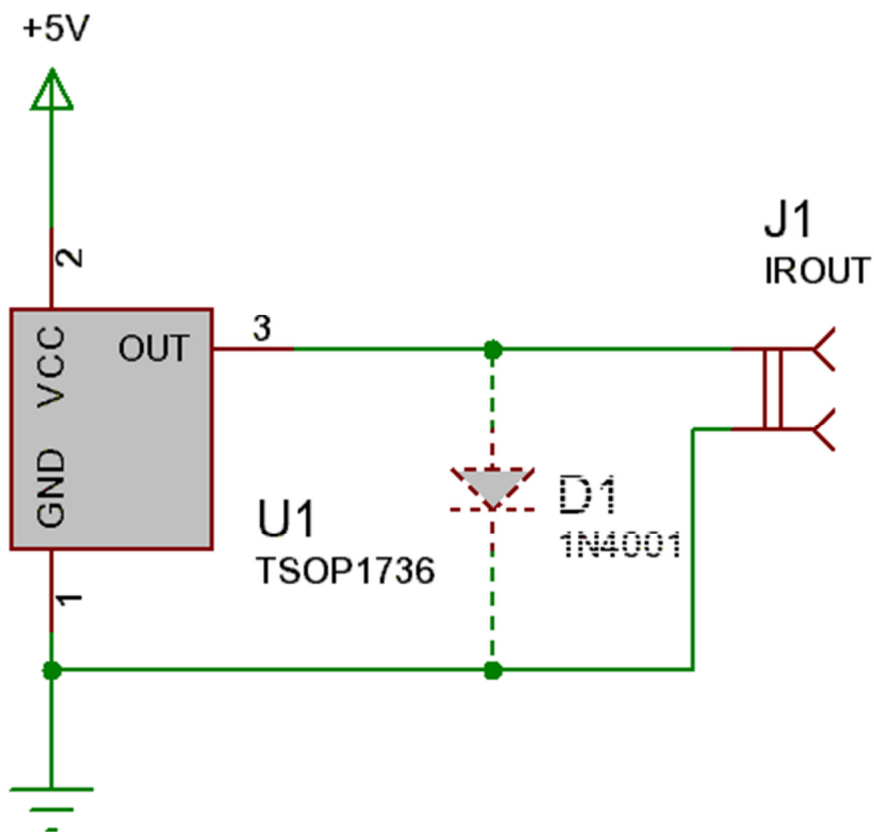
Rysunek 5.14 - Okno główne WinLIRC z ustawionym sygnałem dla helikoptera

Wybrano komendę, która powinna spowodować start helikoptera i jego stopniowe wznoszenie się ku górze. Niestety, po naciśnięciu przycisku „Send Code” helikopter w żaden sposób nie zareagował.

Pierwszą z teorii na temat brak działania był zbyt słaby sygnał z wykorzystanej w pracy diody. Po włączeniu, helikopter przystawiono do nadajnika tak, by odbiornik umieszczony na śmigłowców był w jednej linii z diodą, a odległość między nimi wynosiła 2-3 cm. Niestety zmiany te nie wprowadziły żadnej zmiany – helikopter wciąż nie odpowiadał na wysyłany sygnał.

Kolejnym sprawdzanym elementem były szerokości wysyłanego sygnału. Zmniejszono współczynniki *eps* i *aeps* w pliku konfiguracyjnym. Sytuacja nie uległa jednak zmianie – mimo węższego zakresu długości sygnałów helikopter wciąż nie odpowiadał.

Postanowiono zarejestrować sygnał wysyłany przez nadajnik za pomocą odbiornika. Zważywszy na fakt, że nie można naraz wysyłać i odbierać sygnału przez łącze szeregowe stworzono osobny odbiornik wykorzystujące wejście audio. Schemat nadajnika przedstawiono poniżej:



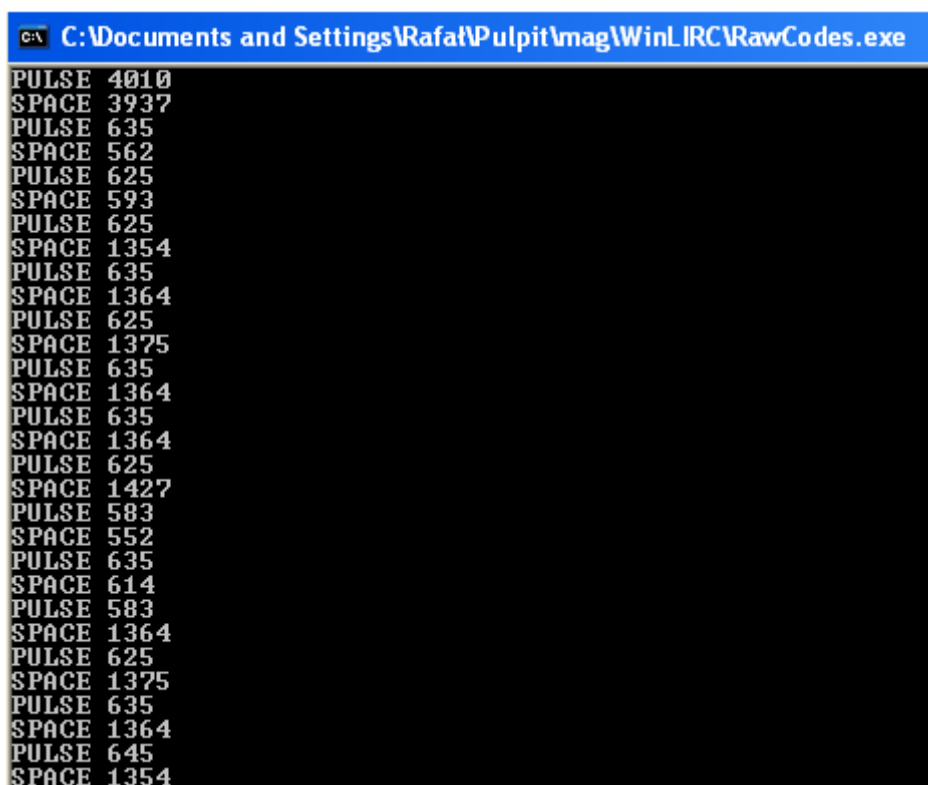
Rysunek 5.15 - Schemat odbiornika podczerwieni.

Źródło: winlirc.sourceforge.net [6]

Tak stworzony odbiornik skonfigurowano w bibliotece WinLIRC i otwarto okno RawCodes. Następnie uruchomiono nadajnik podłączony do łącza szeregowego. Skierowano urządzenia ku sobie i przesłano sygnał taki, jak ten, którego ustawienia widoczne są na rysunku 5.14

Tak skonstruowany odbiornik obserwuje sygnał i analizuje go w inny sposób niż ten skonstruowany wcześniej. W przypadku nowego odbiornik wartości pokazywane przez program RawCodes nie są podawane w mikrosekundach, a w wielokrotnościach czasu próbkowania. Oznacza to dwukrotne zwiększenie wszystkich wartości w porównaniu do odbiornika komunikującego się przez łącze szeregowo.

Zaobserwowano sygnał wysyłany z pilota do helikoptera:



```
C:\> C:\Documents and Settings\Rafał\Pulpit\mag\WinLIRC\RawCodes.exe
PULSE 4010
SPACE 3937
PULSE 635
SPACE 562
PULSE 625
SPACE 593
PULSE 625
SPACE 1354
PULSE 635
SPACE 1364
PULSE 625
SPACE 1375
PULSE 635
SPACE 1364
PULSE 635
SPACE 1364
PULSE 625
SPACE 1427
PULSE 583
SPACE 552
PULSE 635
SPACE 614
PULSE 583
SPACE 1364
PULSE 625
SPACE 1375
PULSE 635
SPACE 1364
PULSE 645
SPACE 1354
```

Rysunek 5.16 - Sygnał wysyłany z pilota widziany przez odbiornik

Na rysunku 5.16 łatwo zaobserwować dwukrotnie większe wartości sygnałów w porównaniu do poprzedniego odbiornika.

Kolejnym krokiem była obserwacja sygnału, jaki wysyłany jest ze skonstruowanego nadajnika. Wyniki przedstawiono na rysunku 5.17.

```
PULSE 3989
SPACE 4083
PULSE 802
SPACE 791
PULSE 812
SPACE 3229
PULSE 572
SPACE 968
PULSE 614
SPACE 885
PULSE 718
SPACE 864
PULSE 802
SPACE 791
PULSE 802
SPACE 708
PULSE 875
SPACE 687
PULSE 510
SPACE 1822
PULSE 802
SPACE 708
PULSE 937
SPACE 614
PULSE 989
SPACE 625
PULSE 979
SPACE 1093
```

Rysunek 5.17 - Sygnał wysyłany z nadajnika widziany przez odbiornik

Zaobserwowano, że niektóre długości przerw są dłuższe niż powinny. Sumaryczna długość całego sygnału nie uległa jednak zmianie. Zmniejszyła się także ilość przerw i pulsów wchodzących w skład sygnału.

Długości zaznaczonych na czerwono przerw nie są przypadkowe. Pierwsza z nich w poprawnym sygnale powinna zostać zastąpiona przerwą po pulsie o długości 812, a także kolejnymi dwoma bitami. W przypadku drugiej zaznaczonej przerwy, w jej miejscu powinna wystąpić przerwa oraz kolejny zakodowany bit.

Niestety po analizie kilkudziesięciu przesłanych sygnałów okazało się, że tylko kilka sygnałów zostało poprawnie wysłanych. Dla porównania stworzono plik konfiguracyjny dla pilota telewizyjnego i zaobserwowano sygnał wysyłany z nadajnika. Sygnał ten wyświetlony został poprawnie, a ponadto uruchomienie go spowodowało zmianę w telewizorze. Oznacza to, że biblioteka, bądź skonstruowany nadajnik ma problemy przy przesyłaniu bardzo krótkich sygnałów.

Postanowiono wydłużyć długości przy kodowaniu bitów tak, aby były poprawnie rozpoznane przez odbiornik.

Minimalne długości, przy których sygnał był tylko sporadycznie źle przesyłany są w granicach 500 mikrosekund. Niestety, przy tak długim kodowaniu niemożliwe jest komunikowanie się z helikopterem według rozpoznanego protokołu.

Podsumowując wybrana metoda komunikacji nie pozwoliła na poprawne sterowanie helikopterem. Możliwości przesyłania sygnału, jakie oferuje łącze szeregowo są niewystarczające, by poprawnie przesyłać sygnał zgodny z wymaganiami czasowymi helikoptera.

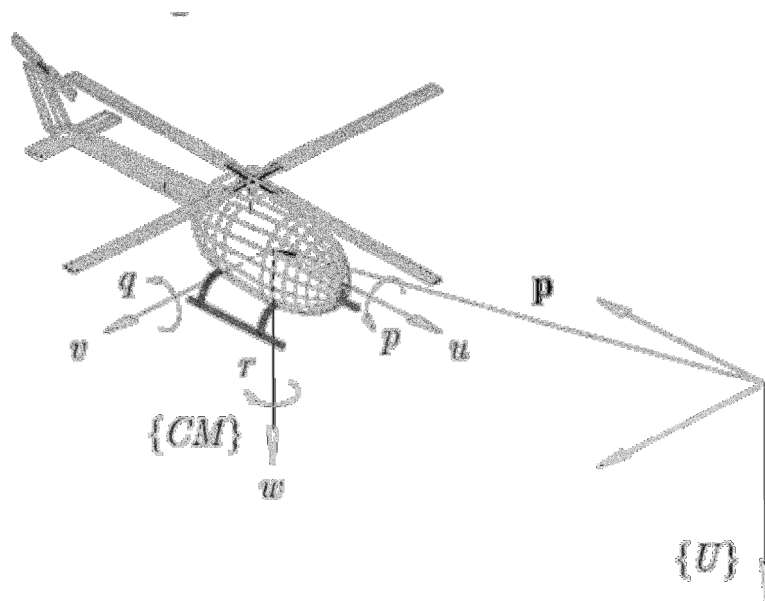
W związku z niepowodzeniem stworzenia poprawnie działającej części nadawczej systemu pozostałą część pracy postanowiono zaimplementować w środowisku MATLAB i przetestować na modelu matematycznym modelującym zachowanie śmigłowca.

6. Model symulacyjny

6.1 Model matematyczny helikoptera

Model matematyczny został wyprowadzony na podstawie opracowań dla helikoptera o budowie standardowej pokazanego na poniższym rysunku. Równania zostały zmodyfikowane tak, aby odpowiadały śmigłowcowi z napędem o budowie Kamowa, czyli takim jakim dysponuje użyty w pracy model. Zmiana ta zmniejsza liczbę stopni swobody helikoptera, a co za tym idzie liczbę zmiennych stanu.

Dodatkowo wprowadzone zostało uproszczenie polegające na tym, że ciąg wytworzony przez tylne śmigło wpływa jedynie na obrót wokół osi poprzecznej (oznaczonej na rysunku jako v), nie wpływając na wartość wypadkową ciągu helikoptera (zmienia jedynie kierunek). Ponadto tylne śmigło tworzy moment obrotowy, który pominięto.



Rysunek 6.1 - Helikopter o standardowej budowie wraz ze zmiennymi i układem współrzędnych

Źródło: Path Following Control for an Autonomous Helicopter [9]

Warto zauważyć, że model opisany na rysunku powyżej posiada nie intuicyjny układ współrzędnych. Największą różnicą jest w takim przypadku wartość siły grawitacji – jest ona dodatnia.

W związku ze wspomnianymi uproszczeniami i założeniami wektor stanu helikoptera jest ośmioelementowy i można go zapisać w poniższy sposób:

$$X(t) = [x(t) \ y(t) \ z(t) \ \alpha(t) \ \beta(t) \ V_{hx}(t) \ V_{hz}(t) \ \omega(t)]^T \quad (6.1)$$

Poniżej przedstawiono stworzony model matematyczny. Został on sporządzony w oparciu o model omówiony w [9].

Algebraiczny zapis równań stanu:

$$\dot{x} = V_{hx} \cdot \cos \alpha \cdot \cos \beta + V_{hz} \cdot \cos \alpha \cdot \sin \beta - \text{zmiana położenia wzdłuż osi x} \quad (6.2)$$

$$\dot{y} = V_{hx} \cdot \sin \alpha \cdot \cos \beta + V_{hz} \cdot \sin \alpha \cdot \sin \beta - \text{zmiana położenia wzdłuż osi y} \quad (6.3)$$

$$\dot{z} = V_{hx} \cdot \sin \beta + V_{hz} \cdot \cos \beta - \text{zmiana położenia wzdłuż osi z} \quad (6.4)$$

$$\dot{\alpha} = \omega_z - \text{zmiana kąta obrotu wokół osi pionowej} \quad (6.5)$$

$$\dot{\beta} = -2 \cdot c_1 \cdot \sin \beta - c_1 \cdot u_3 - \text{zmiana kąta obrotu wokół osi poziomej} \quad (6.6)$$

$$\dot{V}_{hx} = g \cdot \sin \beta + c_4 \cdot u_1 \cdot \sin \beta - \text{zmiana prędkości postępowej helikoptera} \quad (6.7)$$

$$\dot{V}_{hz} = g \cdot \cos \beta - c_4 \cdot u_1 \cdot \cos \beta - \text{zmiana prędkości pionowej helikoptera} \quad (6.8)$$

$$\dot{\omega}_z = -c_2 \cdot \omega_z + c_3 \cdot u_2 - \text{zmiana prędkości kątowej wokół osi pionowej} \quad (6.9)$$

Przyjęte oznaczenia:

u_1 – sterowanie siłą ciągu, $u_1 \in [0,1]$

u_2 – sterowanie skrętne, $u_2 \in [-1,1]$

u_3 – sterowanie postępowe, $u_3 \in [-1,1]$

g – przyspieszenie ziemskie ($9,81 \frac{m}{s^2}$)

Współczynniki c_i charakteryzują konkretny helikopter. Współczynniki te są odpowiedzialne za poszczególne manewry jednostki. Pierwsza z nich odpowiada za prędkość obrotu wokół osi poziomej. Kolejne dwie stałe regulują prędkość maksymalną oraz przyspieszenie obrotu wokół osi pionowej. Ostatnia ze stałych odpowiada za siłę ciągu helikoptera, a co za tym idzie prędkości i przyspieszenia podczas lotu. Domyślne wartości: $c_1 = 1$, $c_2 = 1$, $c_3 = 1$, $c_4 = 20$.

Model uwzględnia trzejelementowe sterowanie – oznacza to, że w porównaniu do helikoptera rzeczywistego przesyłana jest jedna informacja mniej. W przypadku praktycznym przesyłana jest jeszcze informacja o pokrętle stabilizującym. Wynika to z faktu, że model teoretyczny nie uwzględnia zakłóceń oraz niedokładności

powodujących obrót rzeczywistego helikoptera przy zerowym sterowaniu skrotnym – nie ma potrzeby kompensowania zakłóceń za pomocą pokręta.

W modelu matematycznym uwzględniono fakt, że wartość obrotu jednostki wokół osi poziomej jest ograniczona symetrycznie wokół zera. Podstawiając do równania (6.6) skrajne wartości sterowania, czyli odpowiednio -1 i 1, otrzymano sinus kąta, dla którego pochodna jest równa zero. Otrzymano wartość sinusa kąta β odpowiednio -0.5 i 0.5, a co za tym idzie kąt obrotu w modelu jest w zakresie $[-30^0, 30^0]$.

Powyższy model nie uwzględnia zależności między siłą nośną wytwarzaną przez główne śmigła helikoptera, a stanem baterii jednostki. Wartość sterowania, dla którego helikopter pozostanie w zawisie (stan równowagi) można łatwo obliczyć wykorzystując równanie (6.8). Pochodna równania (6.8) przyjmuje wartość zerową w dwóch przypadkach – gdy $\cos\beta$ jest równy zero, bądź gdy zada się odpowiednią wartość sterowania u_1 . Pierwszy z przypadków jest nie osiągalny dla helikoptera, ponieważ oznacza skierowanie dzioba jednostki pionowo w dół. Drugi jest jak najbardziej możliwy, i opisuje sytuację podania sterownia zerowego. Jego wartość obliczono przyrównując do zera równanie (6.8), a następnie dzieląc je przez niezerową wartość $\cos\beta$. W rezultacie otrzymano poniższą zależność:

$$u_1 = \frac{g}{c_4} \quad (6.10)$$

Otrzymane w ten sposób sterowanie bezpośrednio zależy od wartości stałej c_4 . W modelu dokonano także ograniczenia przedziału wartości przyjmowanych przez sterowanie u_1 do przedziału $[0,1]$. Oznacza to, że model będzie mógł ustabilizować lot, jeżeli $\frac{g}{c_4}$ będzie zawierać się w tym właśnie przedziale. Na tej podstawie stwierdzono, że wartość stałej c_4 musi spełniać zależność:

$$c_4 \in (g, \infty) \quad (6.11)$$

6.2 Algorytm rozwiązywania równań różniczkowych

W poprzednim podrozdziale przedstawiono równania różniczkowe opisujące użyty w pracy model helikoptera. Rozwiązując równania w dziedzinie czasu można otrzymać przebiegi czasowe wszystkich zmiennych stanu śmigłowca. Przebiegi te można w prosty sposób otrzymać wykorzystując środowisko MATLAB / Simulink. Jednakże na potrzeby optymalizacji ruchu uzyskanego modelu należało stworzyć algorytm, który oblicza wartości zmiennych stanu w kolejnych chwilach czasowych. W pracy zdecydowano się na implementację algorytmu Rungego - Kutty.

Metoda Rungego - Kutty jest iteracyjną metodą numeryczną rozwiązywania równań różniczkowych zwyczajnych. Zastosowaną odmianą algorytmu jest metoda 4 rzędu. Metoda ta jest charakteryzowana dużą szybkością oraz relatywnie prostą implementacją. Algorytm wyznacza iteracyjnie kolejne wartości rozwiązań równań z określonym krokiem różniczkowania.

Równanie różniczkowe jest postaci: $\dot{X} = rhs(X, u)$, gdzie:

- X_i jest wektorem stanu w i -tej chwili czasowej,
- u_i jest trójelementowym wektorem sterowań w i -tej chwili czasowej,
- funkcja rhs rozwiązuje równania stanu omówione w poprzednim podrozdziale.

W pętli po wszystkich chwilach czasowych t_i , dla kroku różniczkowania h , wyliczane są poniższe cztery współczynniki:

$$k_1 = h \cdot rhs(X_i, u_i) \quad (6.12)$$

$$k_2 = h \cdot rhs\left(X_i + \frac{h}{2}, u_i + \frac{1}{2}k_1\right) \quad (6.13)$$

$$k_3 = h \cdot rhs\left(X_i + \frac{h}{2}, u_i + \frac{1}{2}k_2\right) \quad (6.14)$$

$$k_4 = h \cdot rhs(X_i + h, u_i + k_3) \quad (6.15)$$

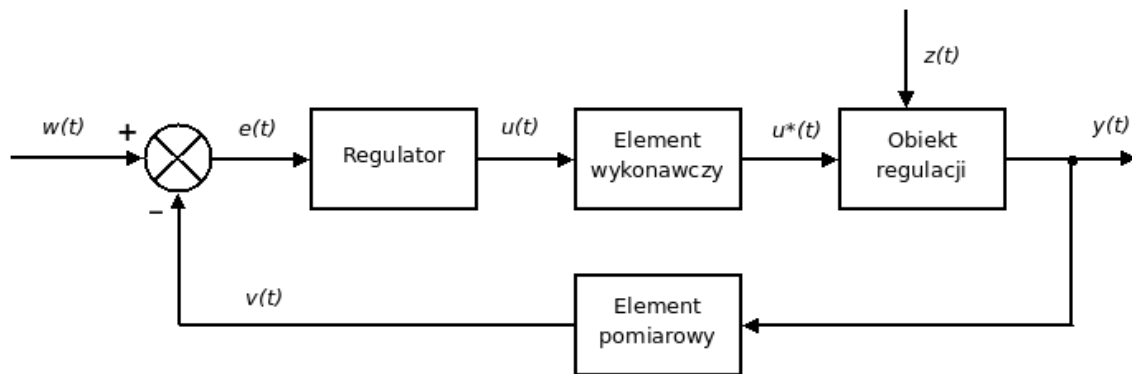
oraz na ich podstawie wartość równania w kolejnych chwilach czasowych:

$$X_{i+1} = X_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (6.16)$$

7. Algorytmy sterowania

7.1 Wstęp

Głównym elementem każdego układu regulacji jest regulator. Ma on na celu na podstawie informacji pozyskanej z wyjścia obiektu wygenerować sterowanie pozwalające na doprowadzenie obiektu do wartości zadanej. Większość regulatorów wykorzystuje uchyb, czyli różnicę pomiędzy wyjściem z obiektu, a wartością zadaną. W takim przypadku zadaniem jest zminimalizowanie tegoż uchybu.



Rysunek 7.1 - Schemat układu regulacji z zaznaczonym regulatorem

Źródło: wikipedia.pl

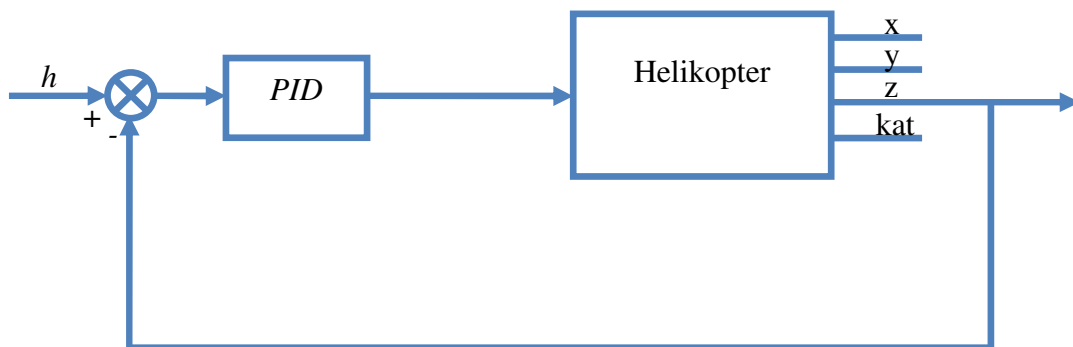
Podstawowe regulatory obliczają wartość sterowania na podstawie aktualnej wartości uchybu. Te bardziej skomplikowane wykorzystują do obliczenia aktualnej wartości sterowania wiedzę o poprzednich wartościach obliczanego sygnału oraz poprzednich wartościach uchybu.

Pracy stworzono algorytmy sterowania, które wykorzystują wiedzę o helikopterze. Algorytm przekazuje do jednostki sterowanie, które ma na celu wykonanie konkretnego zadania, jak na przykład obrót w zadanym kierunku, lot do przodu, czy do tyłu. Procedura zależy od etapu, na jakim aktualnie jest helikopter wybiera do sterowania odpowiedni regulator.

Dla ułatwienia zapisu wartość sygnału sterującego siłą ciągu przyjęto w zakresie od 0 do 1, a pozostałe sygnały sterujące w zakresie od -1 do 1.

7.2 Sterowanie neutralne

Pierwszy z implementowanych algorytmów ma na celu przesłanie do helikoptera sterowania pozwalającego na utrzymanie wysokości h , na której aktualnie się znajduje. Procedura reguluje tylko wartość sterowania wysyłaną do jednostki jako stan lewego drążka, czyli siłę ciągu. Algorytm ten wykorzystuje tylko jeden regulator PID, a schemat regulacji w tym przypadku wygląda następująco:



Rysunek7.2 - Schemat regulacji przy podawaniu sterowania neutralnego

W początkowej wersji pracy w schemacie tym użyto regulatora proporcjonalnego, po którym do wartości sterowania dodawano wartość sterowania zerowego, czyli takiego, które można wyliczyć ze wzoru (6.10). W przypadku rzeczywistym sterowanie to jednak zależne jest od stanu baterii, co wprowadzałoby komplikacje w postaci uchybu ustalonego. Zdecydowano się więc na regulator PID o nastawach:

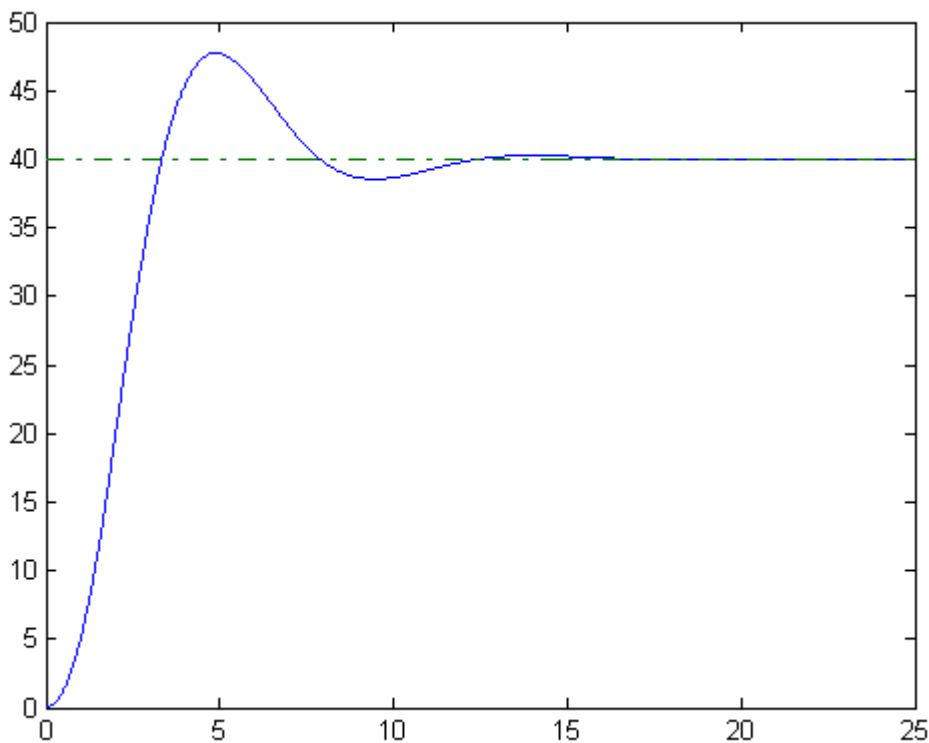
- $P = 0,15$
- $I = 0,1$
- $D = 0,2$

Wartości te znaleziono eksperymentalnie. Warto podkreślić, że tak sformułowany algorytm pozwala nie tylko na zachowanie aktualnej wysokości, ale także jej zmianę. W tym jednak przypadku także występują komplikacje – przy zbyt dużym uchybie występuje zjawisko zwane „windup’em” całkowania. Zjawisko to występuje w układzie sterowania, w którym występuje urządzenie wykonawcze posiadające ograniczenia. W przypadku, gdy zmienna sterowana osiąga wartość graniczną urządzenia wykonawczego pętla sprzężenia zwrotnego zostaje przerwana, ponieważ urządzenie wykonawcze wciąż wysyła wartość maksymalną – niezależnie od sygnału sterującego.

Jeżeli regulator posiada człon całkujący – jak ten użyty w pracy – uchyb nadal będzie całkowany. Oznacza to, że wyrażenie całkujące osiąga szybko bardzo wysoką wartość.

W celu zniwelowania tegoż efektu należy w regulatorze zastosować tzw. "anti-windup". Polega on na odejmowaniu od wyrażenia całkującego różnicy pomiędzy wartością sterowania, a ograniczeniem. W środowisku MATLAB/Simulink „anty windup” został zaimplementowany wewnątrz bloku regulatora PID.

Schemat regulacji z rysunku 7.2 zaimplementowano w środowisku MATLAB/Simulink. Poniżej przedstawiono przebieg zmiennej z helikoptera startującego z poziomu 0, który ma za zadanie osiągnięcie wysokości równej 40.

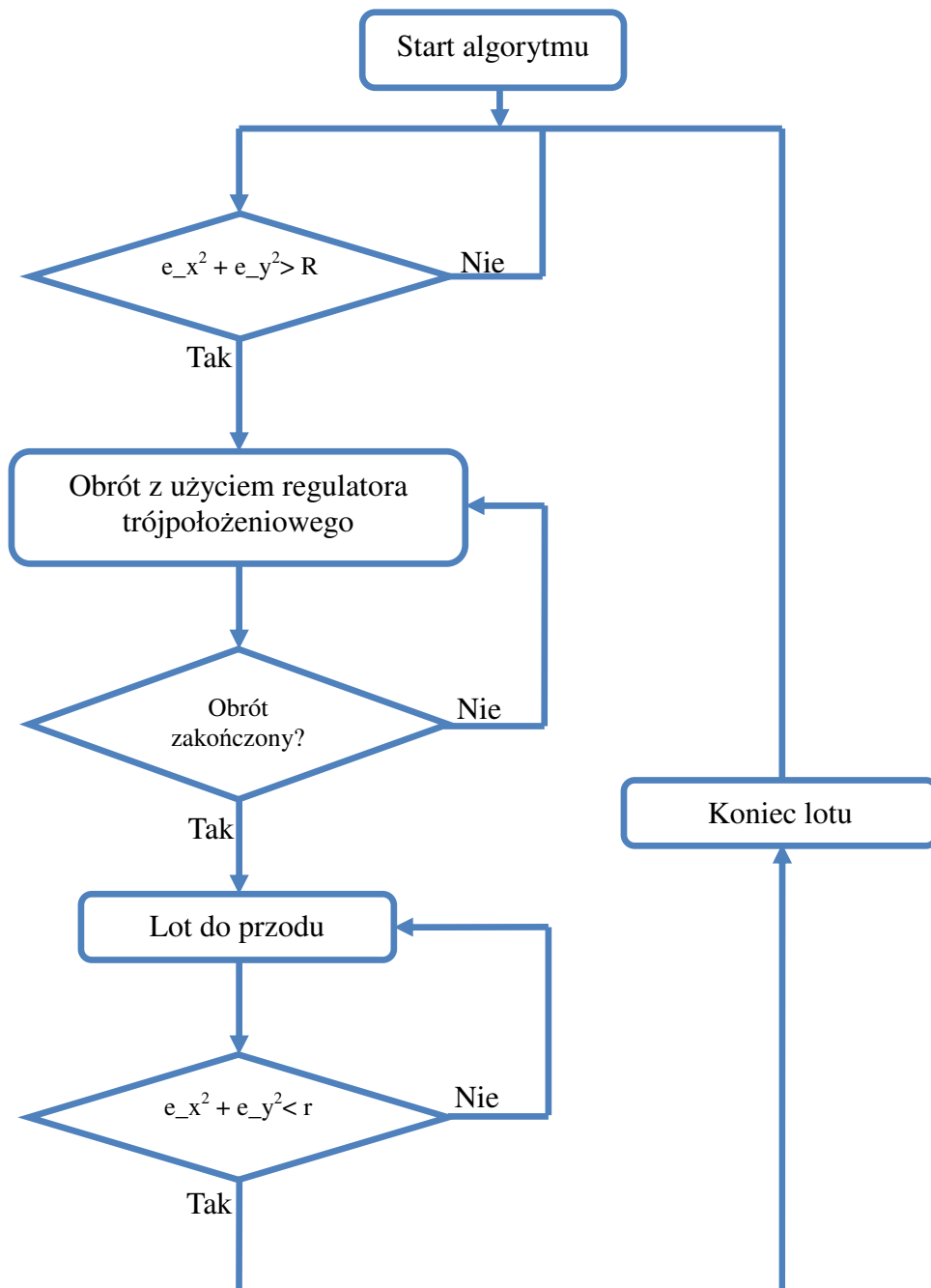


Rysunek 7.3 – Przebieg wysokości helikoptera w czasie

W teorii śmigłowiec nigdy nie powinien opuścić otoczenia zadanego punktu. W przypadku rzeczywistym jest jednak inaczej – zakłócenia powstałe na skutek niedokładności układu, wiatru oraz problemów z przesyłaniem sterowania powodują przesunięcie jednostki od zadanego punktu. Wymagany jest wówczas bardziej skomplikowany algorytm sterujący.

7.3 Algorytm stabilizacji w punkcie

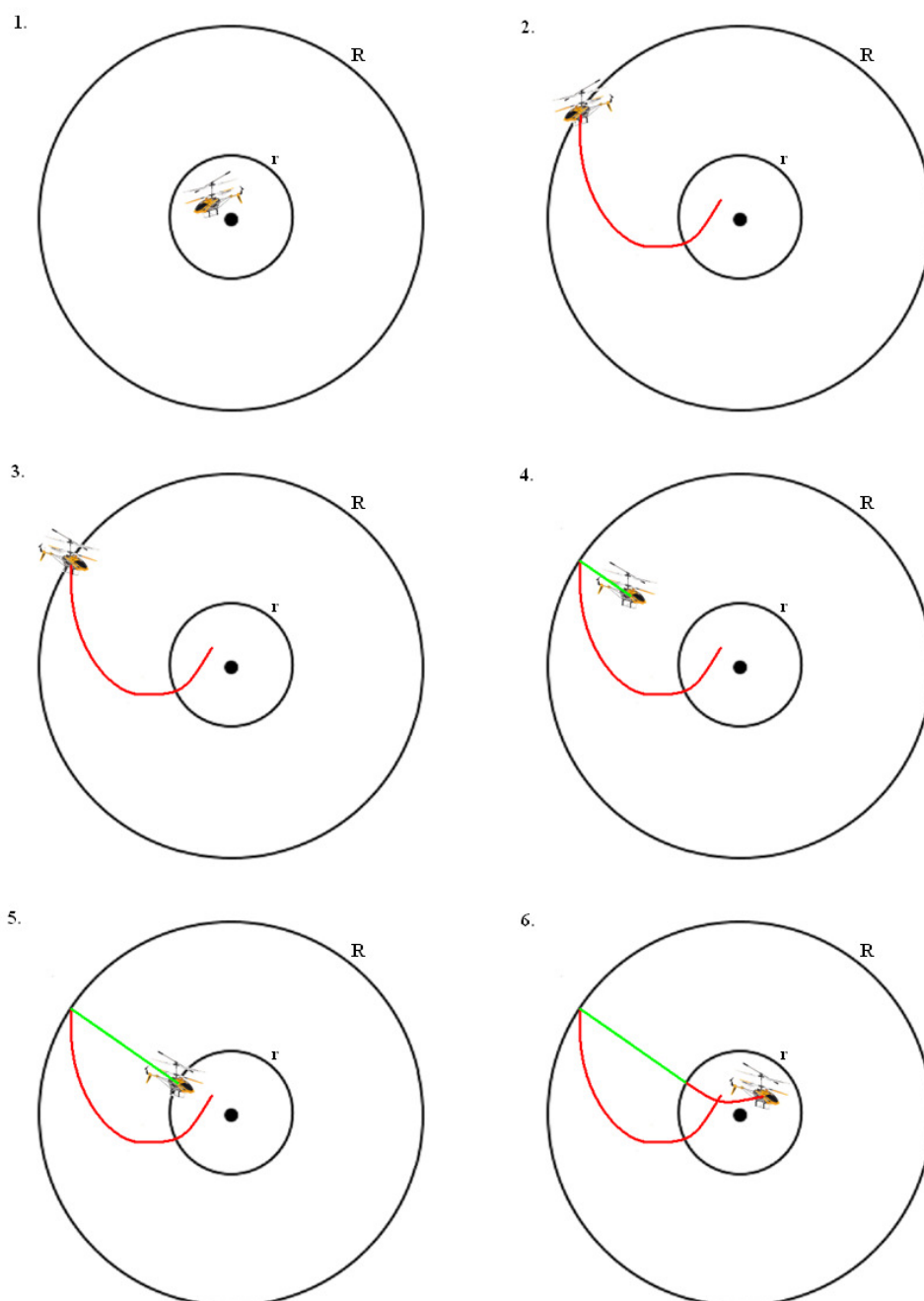
Kolejny algorytm ma na celu stabilizację helikoptera w punkcie aktualnego położenia wykorzystując procedurę utworzoną w poprzednim podrozdziale. Jak wspomniano podawanie sygnału neutralnego nie zapewnia utrzymania helikoptera w jednym punkcie. Jeżeli jednostka znajdzie się odpowiednio daleko od zadanego punktu należy za pomocą sygnału sterującego dokonać obrotu śmigłowca, po którym nastąpi lot w pobliże punktu pierwotnego położenia.



Rysunek 7.4 – Schemat algorytmu powrotu do punktu zadanego

Przedstawiony algorytm pozwala na doprowadzenie helikoptera z powrotem do zadanego punktu. Można zauważyć, że algorytm wykorzystuje jedynie obrót oraz ruch postępowy śmigłowca. Do poprawnego działania niezbędne jest równoległe podawanie sterowania neutralnego według omówionej w poprzednim podrozdziale procedury.

Działanie algorytmu zaprezentowano na poniższych grafikach:



Rysunek 7.5 – Graficzna prezentacja kolejnych kroków algorytmu

Na grafikach kolorem czerwonym oznaczono drogę helikoptera przy sterowaniu neutralnym, a zielonym ruch wynikający ze sterowania obliczanego w ramach omawianego algorytmu.

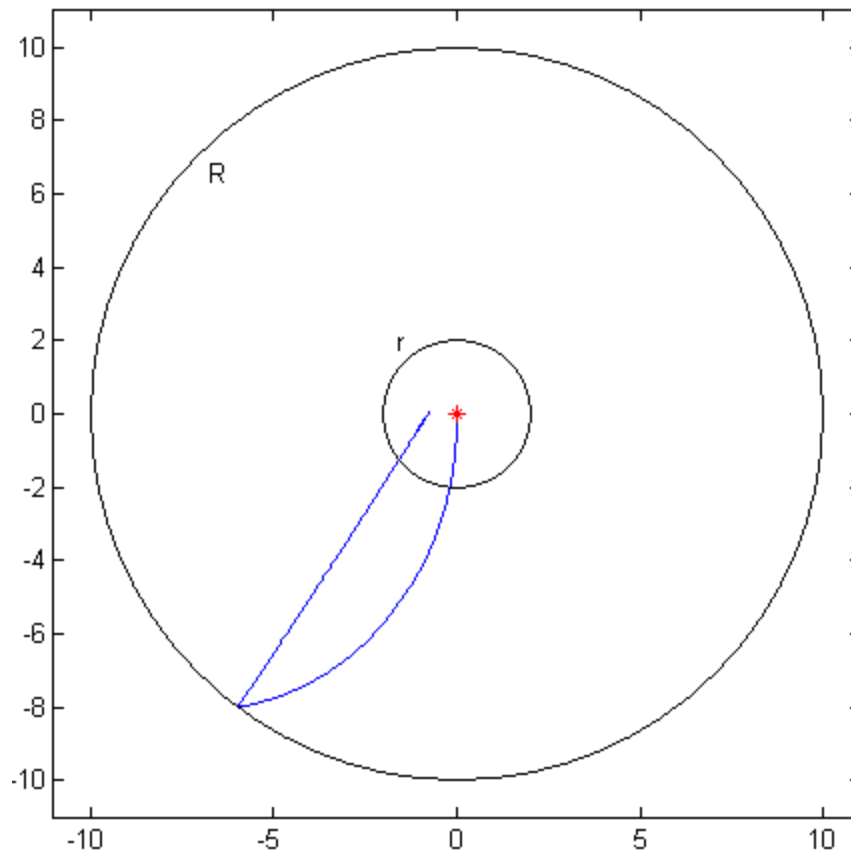
Opis przedstawionych kroków algorytmu:

1. Początek działania algorytmu – punkt, w którym znajduje się helikopter zapamiętany jest jako punkt zadany.
2. Po odpowiednio długim przesyłaniu do helikoptera sterowania neutralnego śmigłowiec oddalił się od punktu zadanego na odległość większą od R
3. Nastąpił obrót helikoptera tak, aby śmigłowiec był skierowany w kierunku punktu docelowego
4. Po obrocie helikopter rozpoczął lot w kierunku punktu docelowego. Zbliżając się do odległości r rozpoczyna hamowanie.
5. Śmigłowiec znalazł się w odległości mniejszej niż r od punktu docelowego – lot został zakończony, a zaczęto przysyłać sterowanie neutralne
6. W wyniku przesyłania sterowania neutralnego helikopter znów zaczyna oscylować wokół punktu docelowego.

Jak widać w algorytmie ważną rolę pełnią parametry R i r , czyli promienie odpowiednio dużego i małego koła przedstawionego na grafikach. Duże koło odpowiada za odległość, którą uznano za zbyt dużą i wymaga się powrotu w bliższe otoczenie punktu. Małe koło prezentuje otoczenie punktu docelowego, które uznano za zadowalające. Jego odpowiednia wielkość pozwala na zniwelowanie niedokładności układu stereowizyjnego, czy też wykonanego przez helikopter obrotu.

Do symulacyjnego sprawdzenia poprawności działania algorytmu należało wprowadzić pewne zmiany w modelu matematycznym tak, aby helikopter mimo podawania sterowania neutralnego opuścił punkt zadany. W tym celu w prosty sposób zasymulowano działanie wiatru – dodano do wartości pochodnych zmiennych x i y pewną niezerową wartość.

Wynikiem tych zabiegów oraz działania algorytmu jest poniższa trajektoria helikoptera:

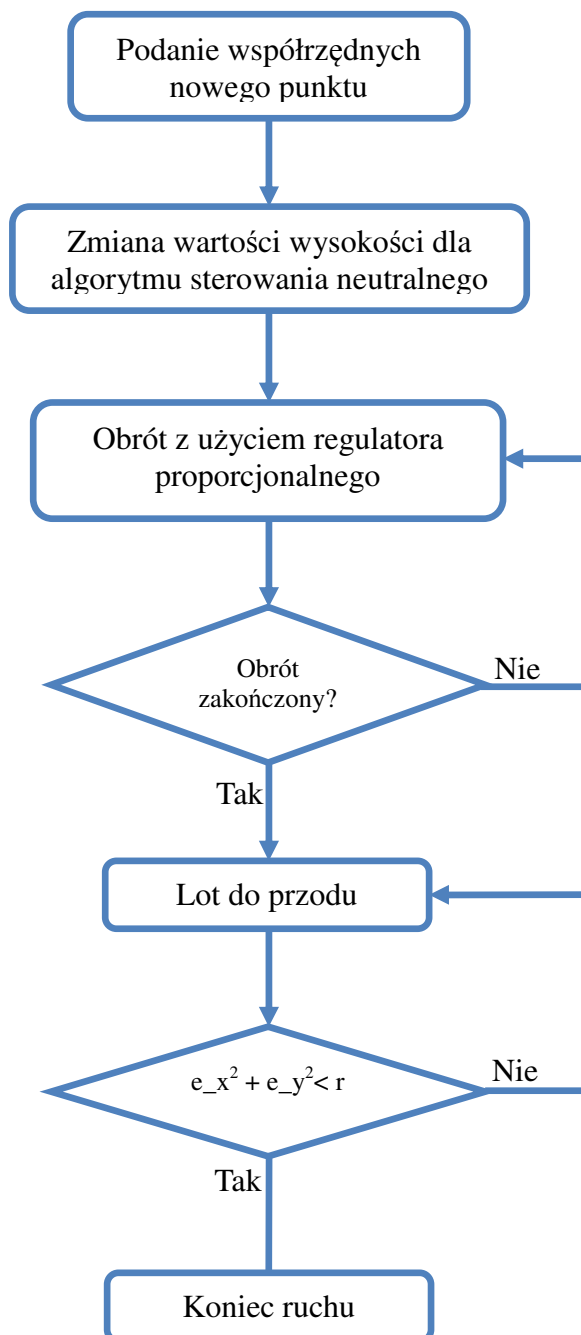


Rysunek 7.6 – Trajektoria ruchu helikoptera (płaszczyzna xy) z zaznaczeniem kół o promieniu r i R

Powyższy rysunek pokazuje poprawne działanie stworzonego algorytmu – dodatkowo wzięto pod uwagę fakt, że model wykazuje mniejsze samohamownie niż obiekt rzeczywisty – w tym celu hamowanie lotu odbyło się za pomocą regulatora. W przypadku rzeczywistym do hamowania lotu wystarczy za wartość sterowania postępowego przyjąć zero.

7.4 Algorytm lotu do punktu

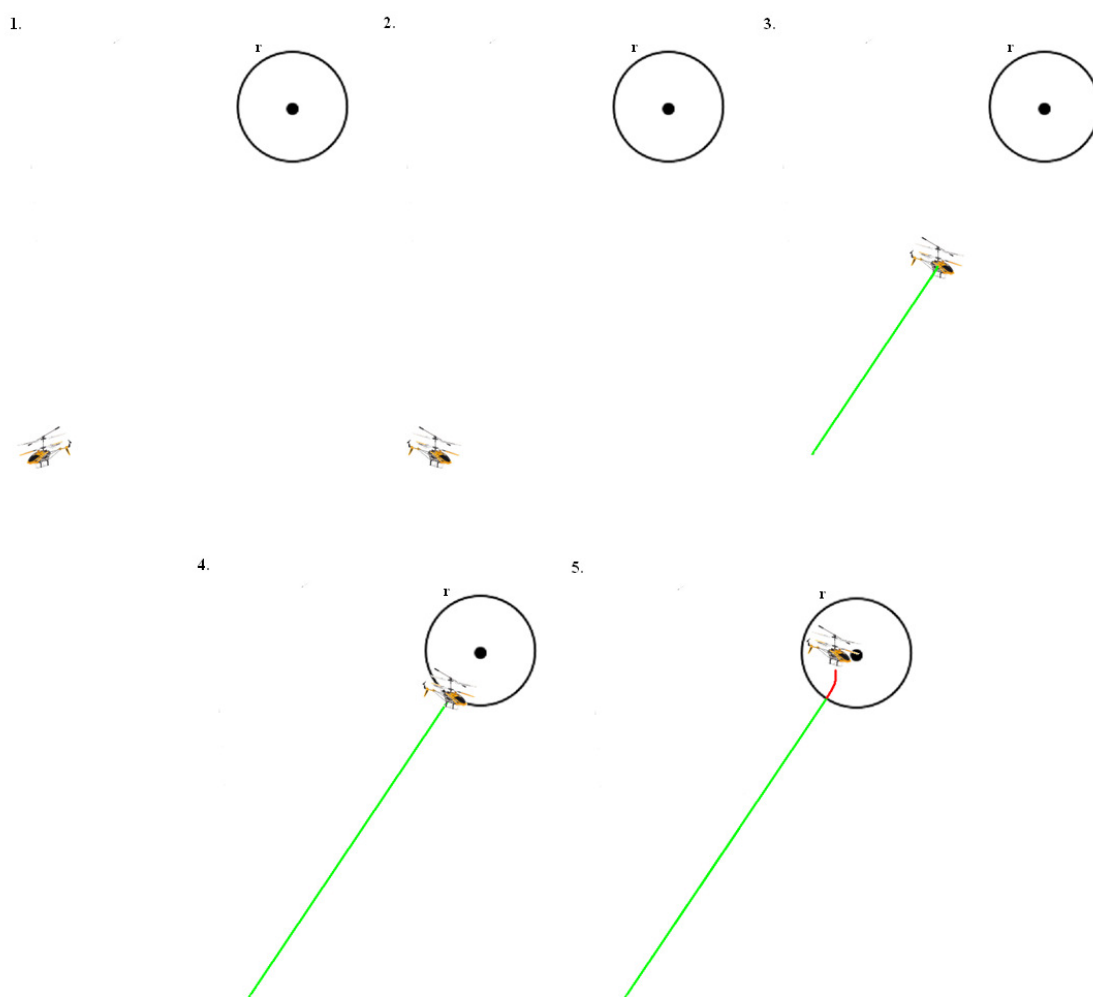
Kolejny z zaproponowanych algorytmów pozwala na przemieszczenie się helikoptera z jego aktualnego położenia do zadanego punktu w przestrzeni. Procedura ta jest bardzo podobna do tej przedstawionej w poprzednim podrozdziale. Przypadek ten różni się od poprzedniego możliwą zmianą wysokości. Jednak odpowiednio stworzony algorytm poszukiwania sterowania zerowego zapewnia dostosowanie wartości siły ciągu do nowego poziomu.



Rysunek 7.7 – Schemat blokowy algorytmu lotu do punktu

Powyższy algorytm na podstawie współrzędnych nowego punktu ustala nową wysokość dla algorytmu poszukiwania sterowania zerowego oraz oblicza kąt obrotu śmigłowca tak, by ten był skierowany w kierunku punktu docelowego. Po wykonaniu obrotu następuje ruch postępowy, który zostaje zakończony w odpowiednio małej odległości od wartości zadanych. Zarówno podczas obrotu, jak i następującego po nim lotu można zaobserwować wznoszenie się helikoptera dzięki działającemu równolegle algorytmowi sterowania neutralnego.

Działanie algorytmu zaprezentowano na poniższych grafikach:



Rysunek 7.8 – Graficzna prezentacja kolejnych kroków algorytmu

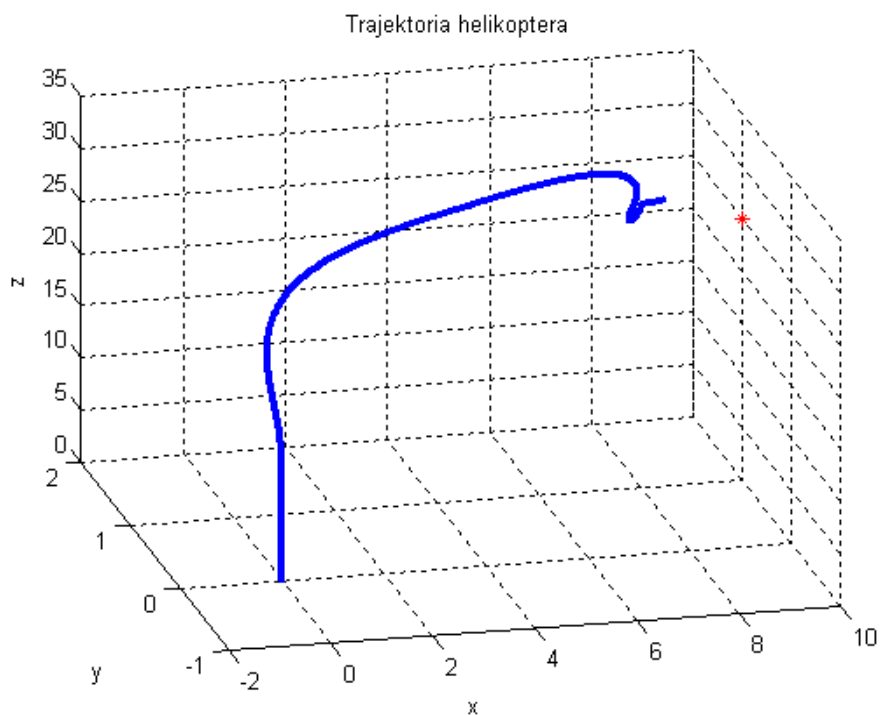
Na grafikach kolorem czerwonym oznaczono drogę helikoptera przy sterowaniu neutralnym, a zielonym ruch wynikający ze sterowania obliczanego w ramach omawianego algorytmu.

Opis przedstawionych kroków algorytmu:

1. Początek działania algorytmu – wysokość punktu zadanego jest podawana do algorytmu sterowania neutralnego oraz obliczany jest kąt obrotu
2. Helikopter zakończył obrót w kierunku punktu docelowego
3. Rozpoczęcie lotu postępowego
4. Zakończenie lotu postępowego po zbliżeniu się śmigłowca do punktu zadanego na odległość mniejszą niż r
5. W wyniku przesyłania sterowania neutralnego helikopter znów zaczyna oscylować wokół punktu docelowego.

Koło o promieniu r widoczne na grafikach prezentuje otoczenie punktu docelowego które uznano za zadowalające. Jego odpowiednia wielkość pozwala na zniwelowanie niedokładności układu stereowizyjnego, czy też wykonanego przez helikopter obrotu.

Algorytm przetestowano na modelu symulacyjnym w ten sam sposób, co algorytm stabilizacji w punkcie. Efekt widoczny na rysunku 7.9.



Rysunek 7.9 - Trajektoria helikoptera przy locie do punktu [10,1,25]

Pozornie może się wydawać, że punkt końcowy znajduje się daleko od punktu zadanego oznaczonego na czerwono. Należy jednak pamiętać o tym, że algorytm kończy pracę, gdy helikopter znajduje się w kole o promieniu r i środku w punkcie zadanym. W powyższym przypadku punktem końcowym działania algorytmu jest punkt [9.055, 1.31, 25.13]. Dla wartości r równej 2 punkt ten znajduje się w kole, co oznacza, że algorytm zadziałał poprawnie.

8. Algorytm sterowania optymalnego

8.1 Sformułowanie problemu optymalizacji i jego analiza

Pierwszym etapem formułowania problemu jest stworzenie jego postaci pierwotnej. W pracy postawione zostało zadanie minimalizacji pierwotnego wskaźnika jakości przy spełnionych wymaganych warunkach końcowych. Zadanie polega na osiągnięciu przez helikopter zerowego punktu w przestrzeni kartezyjskiej oraz zatrzymaniu się i stabilizacji w tym punkcie. Położenie początkowe helikoptera oraz horyzont czasu sterowania T są znane.

Wskaźnik jakości został uzależniony od napięcia, które z pewną wagą jest proporcjonalne do sterowania jawnie występującego w równaniach stanu. Ze względu na niewielkie nieliniowości wykazywane przez układ (zgodnie z zapewnieniami producenta), moc pobieraną z kondensatora można wyrazić, jako proporcjonalną do kwadratu napięcia, co niesie za sobą proporcjonalność do kwadratu sterowania. Poszczególne elementy wskaźnika jakości mnożone są przez różne współczynniki, ponieważ proporcje poboru mocy są różne dla różnych składowych sterowania.

Innym możliwym wskaźnikiem jakości w omawianym problemie mógłby być całkowity ładunek elektryczny pobrany z kondensatora, jednak ostatecznie wybrana została pierwsza z wymienionych możliwości. Pierwotny wskaźnik jakości jest więc postaci:

$$Q(u) = \frac{1}{2} \int_0^T u(t)^T c_Q u(t) dt, \quad (8.1)$$

gdzie:

$$u(t) = [u_1(t) u_2(t) u_3(t)]$$

$$c_Q = \begin{bmatrix} c_{Q1} & 0 & 0 \\ 0 & c_{Q2} & 0 \\ 0 & 0 & c_{Q3} \end{bmatrix}$$

Wprowadzono także ograniczenia końcowe:

$$x(T) = y(T) = z(T) = V_{hx}(T) = V_{hz}(T) = \omega(T) = 0$$

Kolejnym krokiem było zmodyfikowanie współczynnika jakości tak, by ten uwzględniał funkcję kary.

Zmodyfikowany wskaźnik jakości:

$$Q_p(u) = Q(u) + \frac{1}{2}\rho X(T)^T W X(T), \quad (8.2)$$

gdzie:

ρ - współczynnik funkcji kary

$$W = \begin{bmatrix} 1 & 0 & . & . & . & . & . & 0 \\ 0 & 1 & 0 & . & . & . & . & . \\ . & 0 & 1 & 0 & . & . & . & . \\ . & . & 0 & 0 & 0 & . & . & . \\ . & . & . & 0 & 0 & 0 & . & . \\ . & . & . & . & 0 & 1 & 0 & . \\ . & . & . & . & . & 0 & 1 & 0 \\ 0 & . & . & . & . & . & 0 & 1 \end{bmatrix} - \text{macierz ograniczeń końcowych}$$

Jak łatwo zauważyć ograniczenia końcowe nie dotyczą zmiennej α oraz β . Są to zmienne zależne od orientacji helikoptera – odpowiednio jest to kąt obrotu wokół pionowej oraz poziomej osi. Żądanie wyzerowania wartości pierwszej z nich w punkcie końcowym byłoby jednoznaczne z żądaniem ustawienia się helikoptera zawsze w takiej samej orientacji, co w omawianym zadaniu sterowania nie jest konieczne. Druga z tych zmiennych w przypadku żądania zatrzymania się w zadanym punkcie końcowym i tak zostanie wyzerowana, ponieważ od jej wartości zależy ruch postępowy helikoptera.

Następnym krokiem była analiza sformułowanego problemu za pomocą zasady maksimum Pontriagina. Pierwszym elementem tejże analizy było zdefiniowanie Hamiltonianu dla omawianego problemu. Hamiltonian zdefiniowany jako funkcja $H: \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ w pracy przyjmuje postać:

$$H(\psi(t), X(t), u(t)) = \psi^T(t) \cdot f(X(t), u(t)) - \frac{1}{2}u(t)^T c_Q u(t), \quad (8.3)$$

gdzie ψ jest funkcją sprzężoną $\psi: [0, T] \rightarrow \mathbb{R}^n$

Twierdzenie 1 – zasada maksimum Pontriagina

Założmy, że u^* jest sterowaniem optymalnym, X^* wygenerowaną przez nie optymalną trajektorią stanu, a ψ^* odpowiednią optymalną trajektorią sprzężoną. Zbiór D jest zbiorem sterowań dopuszczalnych. Przy tych założeniach, dla wszystkich $t \in [0, T)$ zachodzi warunek:

$$H(\psi^*(t), X^*(t), u^*(t)) \geq H(\psi^*(t), X^*(t), v) \quad \forall v \in D$$

to znaczy hamiltonian jest maksymalizowany przez sterowanie optymalne.

W przypadku, gdy sterowanie optymalne leży wewnątrz zbioru dopuszczalnego można je wyznaczyć przyrównując pochodne cząstkowe Hamiltonianu względem sterowań do zera:

$$\frac{\partial H}{\partial u_1} = \psi_6 \cdot c_4 \cdot \sin \beta - \psi_7 \cdot c_4 \cdot \cos \beta - u_1 \cdot c_{Q1} = 0 \quad (8.4)$$

$$\frac{\partial H}{\partial u_2} = \psi_8 \cdot c_3 - u_2 \cdot c_{Q2} = 0 \quad (8.5)$$

$$\frac{\partial H}{\partial u_3} = \psi_5 \cdot c_1 - u_3 \cdot c_{Q3} = 0 \quad (8.6)$$

Po przekształceniach otrzymano:

$$u_1^* = \frac{1}{c_{Q1}} (\psi_6 \cdot c_4 \cdot \sin \beta - \psi_7 \cdot c_4 \cdot \cos \beta) \quad (8.7)$$

$$u_2^* = \frac{1}{c_{Q2}} (\psi_8 \cdot c_3) \quad (8.8)$$

$$u_3^* = \frac{1}{c_{Q3}} (\psi_5 \cdot c_1) \quad (8.9)$$

Jeżeli jednak zachodzi sytuacja, w której sterowanie znajduje się poza zbiorem dopuszczalnym to wtedy element wektora sterowań wykraczający poza zakres przyjmuje jedną z wartości brzegowych. Wybierana jest ta wartość brzegowa, dla której wartość hamiltonianu jest większa.

Równania (8.7 – 8.9) jednoznacznie określają poszukiwane sterowanie optymalne. Do ich rozwiązania należy jednak wyznaczyć funkcję sprzężoną ψ . Funkcja ta jest definiowana jako rozwiązanie równania różniczkowego przedstawionego poniżej:

$$\dot{\psi}(t) = -\nabla_x H(\psi(t), X(t), u(t)), \quad t \in [0, T] \quad (8.10)$$

z warunkiem końcowym:

$$\psi(T) = -\nabla_{X(T)} q(X(T), u) \quad (8.11)$$

Równania sprzężone dla omawianego modelu mają postać:

$$\dot{\psi}_1(t) = -\frac{\partial H}{\partial x} = 0 \quad (8.12)$$

$$\dot{\psi}_2(t) = -\frac{\partial H}{\partial y} = 0 \quad (8.13)$$

$$\dot{\psi}_3(t) = -\frac{\partial H}{\partial z} = 0 \quad (8.14)$$

$$\dot{\psi}_4(t) = -\frac{\partial H}{\partial \alpha} = (\psi_1 \cdot \sin \alpha - \psi_2 \cdot \cos \alpha) \cdot (V_{hx} \cdot \cos \beta - V_{hz} \cdot \sin \beta) \quad (8.15)$$

$$\dot{\psi}_5(t) = -\frac{\partial H}{\partial \beta} = (\psi_1 \cdot \cos \alpha + \psi_2 \cdot \sin \alpha) \cdot (V_{hx} \cdot \sin \beta - V_{hz} \cdot \cos \beta) + \psi_3 \cdot \sin \beta \cdot \\ \cdot V_{hz} + \psi_5 \cdot \cos \beta \cdot 2c_1 - \psi_6 \cdot \cos \beta \cdot (g + c_4 \cdot u_1) + \psi_6 \cdot \sin \beta \cdot (g - c_4 \cdot u_1) \quad (8.16)$$

$$\dot{\psi}_6(t) = -\frac{\partial H}{\partial v_{hx}} = -\psi_1 \cdot \cos \alpha \cdot \cos \beta - \psi_2 \cdot \sin \alpha \cdot \cos \beta - \psi_3 \cdot \sin \beta \quad (8.17)$$

$$\dot{\psi}_7(t) = -\frac{\partial H}{\partial v_{hz}} = -\psi_1 \cdot \cos \alpha \cdot \sin \beta - \psi_2 \cdot \sin \alpha \cdot \sin \beta - \psi_3 \cdot \cos \beta \quad (8.18)$$

$$\dot{\psi}_8(t) = -\frac{\partial H}{\partial \omega_z} = -\psi_4 + \psi_8 \cdot c_2 \quad (8.19)$$

Warunek końcowy:

$$\psi(T) = -\frac{\partial q(X(T), u)}{\partial X(T)} = -\rho W X(T) \quad (8.20)$$

Równania sprzężone występujące w hamiltonianie są równaniami różniczkowymi ze znaną, zadaną wartością końcową. Z tego powodu rozwiązanie ich w sposób standardowy taki, jak w przypadku równań stanu jest niemożliwy. Konieczne jest zaimplementowanie metody rozwiązującej równania różniczkowe z czasem odwrotnym („wstecz”). Metoda jest analogiczna do standardowej metody rozwiązywania „w przód”.

W celu wyznaczenia punktów węzłowych oraz końcowej wartości wektora stanu rozwiązane zostają równania stanu metodą RK4 w wersji standardowej. Wyznaczone wartości w poszczególnych punktach są zapamiętywane, aby zredukować ilość

późniejszych obliczeń. Dla kolejnych chwil czasowych, poczynając od czasu końca symulacji a kończąc na czasie zerowym, wyliczane są wartości równań stanu oraz z ich wykorzystaniem wartości równań sprzężonych.

Wprowadzono następujące oznaczenia:

$\dot{z} = f(z, u)$, $z = [X, \psi]$, gdzie:

z_i - wektor w i -tej chwili czasowej złożony z wektora stanu i wektora zmiennych sprzężonych

u_i - trójelementowy wektor sterowań w i -tej chwili czasowej

f – funkcja rozwiązująca równania stanu i równania sprzężone omówione wcześniej

W pętli po wszystkich chwilach czasowych t_i , dla kroku całkowania h wyliczane są poniższe współczynniki:

$$k_1 = f(z_i, u_i) \quad (8.21)$$

$$k_2 = f\left(z_i - \frac{h}{2}k_1, u_i\right) \quad (8.22)$$

$$k_3 = f\left(z_i - \frac{h}{2}k_2, u_i\right) \quad (8.23)$$

$$k_4 = f(z_i - h \cdot k_3, u_i) \quad (8.24)$$

Następnie na ich podstawie obliczana jest wartość zmiennej z w poprzedniej chwili czasowej:

$$z_{i-1} = z_i - \frac{h}{3}(k_2 + k_3) - \frac{h}{6}(k_1 + k_4) \quad (8.25)$$

Wektor z_{i-1} daje informacje o wartościach stanu w chwili $i-1$, jednakże są one obciążone większym błędem numerycznym niż te uzyskane z rozwiązania równań stanu „w przód”. Dlatego do wyliczania wartości zmiennych sprzężonych brane są pod uwagę zapamiętane wartości zmiennych stanu wyznaczone wcześniej.

Ostatnim krokiem było sformułowanie indukowanego wielowymiarowego problemu optymalizacji oraz obliczenie jego gradientu. W pracy do aproksymacji sterowania została wybrana postać schodkowa. W rozumieniu matematycznym takie sterowanie

jest funkcją przedziałami stałą. Liczba przedziałów jest dowolna i zostaje zdefiniowana przed uruchomieniem algorytmu optymalizacyjnego. Liczba przedziałów jest taka sama dla wszystkich trzech sterowań. Oznacza to, że liczba zmiennych decyzyjnych jest równa liczbie przedziałów pomnożonej przez trzy.

Oznaczenia:

n - liczba przedziałów

dykretyzacja czasu: $0 = t_1 < t_2 < \dots < t_{n+1} = T$

sterowanie schodkowe: $u_k(t) = u_{ki}$, $t \in [t_i, t_{i+1})$, $i = 1, \dots, n$, $k = 1, 2, 3$, $u_k(T) = u_{k,n}$

$u_{11}, u_{12}, \dots, u_{1n}$ – zmienne decyzyjne jednoznacznie określające sterowanie u_1

$u_{21}, u_{22}, \dots, u_{2n}$ – zmienne decyzyjne jednoznacznie określające sterowanie u_2

$u_{31}, u_{32}, \dots, u_{3n}$ – zmienne decyzyjne jednoznacznie określające sterowanie u_3

ograniczenia:

$u_{1i} \in [0, 1]$ dla $i = 1, 2, \dots, n$

$u_{2i}, u_{3i} \in [-1, 1]$ dla $i = 1, 2, \dots, n$

Biorąc pod uwagę charakter wybranego sterowania uproszczono zdefiniowany wcześniej zmodyfikowany wskaźnik jakości do postaci zależnej od zmiennych decyzyjnych:

$$Q_{\rho D}(u) = \frac{1}{2} \rho X(T)^T W X(T) + \frac{1}{2} \sum_{i=1}^n (c_{Q1} u_{1i}^2 + c_{Q2} u_{2i}^2 + c_{Q3} u_{3i}^2) \Delta t, \text{ gdzie:} \quad (8.26)$$

$$\Delta t = t_{i+1} - t_i$$

Pochodna wskaźnika jakości względem i -tego wektora zmiennych decyzyjnych:

$$G_{ik} = - \int_{t_i}^{t_{i+1}} \nabla_{u_k} H(\psi, X, u_{1i}, u_{2i}, u_{3i}) dt, \quad i = 1, 2, \dots, n; \quad k = 1, 2, 3 \quad (8.27)$$

W pracy gradient hamiltonianu jest postaci:

$$\nabla_{u_1} H(\psi, X, u_{1i}, u_{2i}, u_{3i}) = \psi_6 \cdot c_4 \cdot \sin \beta - \psi_7 \cdot c_4 \cdot \cos \beta - c_{Q1} \cdot u_{1i} \quad (8.28)$$

$$\nabla_{u_2} H(\psi, X, u_{1i}, u_{2i}, u_{3i}) = \psi_8 \cdot c_3 - c_{Q2} \cdot u_{2i} \quad (8.29)$$

$$\nabla_{u_3} H(\psi, X, u_{1i}, u_{2i}, u_{3i}) = -\psi_5 \cdot c_1 - c_{Q3} \cdot u_{3i} \quad (8.30)$$

Następnie obliczono elementy macierzy G:

$$G_{i1} = - \int_{t_i}^{t_{i+1}} (\psi_6 \cdot c_4 \cdot \sin \beta - \psi_7 \cdot c_4 \cdot \cos \beta) dt + \Delta t \cdot c_{Q1} + u_{1i} \quad (8.31)$$

$$G_{i2} = - \int_{t_i}^{t_{i+1}} (\psi_8 \cdot c_3) dt + \Delta t \cdot c_{Q2} + u_{2i} \quad (8.32)$$

$$G_{i3} = - \int_{t_i}^{t_{i+1}} (-\psi_5 \cdot c_1) dt + \Delta t \cdot c_{Q3} + u_{3i} \quad (8.33)$$

Co można zapisać następująco:

$$G_{ik} = \Delta t \cdot c_Q \cdot u_{ki} + y_{ik}(t_i), \quad (8.34)$$

gdzie:

$$y_{i1}(t_i) = - \int_{t_i}^{t_{i+1}} (\psi_6 \cdot c_4 \cdot \sin \beta - \psi_7 \cdot c_4 \cdot \cos \beta) dt \quad , y_{i1}(t_{i+1}) = 0$$

$$y_{i2}(t_i) = - \int_{t_i}^{t_{i+1}} (\psi_8 \cdot c_3) dt \quad , y_{i2}(t_{i+1}) = 0$$

$$y_{i3}(t_i) = - \int_{t_i}^{t_{i+1}} (-\psi_5 \cdot c_1) dt \quad , y_{i3}(t_{i+1}) = 0$$

8.2 Algorytm optymalizacji gradientowej

W pracy do rozwiązywania omówionego problemu wykorzystano algorytm Broydena – Fletchera – Goldfarba – Shanno, czyli w skrócie BFGS. Algorytm ten wykorzystuje się do numerycznego rozwiązywania nieliniowych problemów optymalizacji. W literaturze można spotkać kilka wersji tegoż algorytmu – w pracy wykorzystano wersję z odnową i progami. Poszczególne kroki przedstawiono poniżej.

W celu uproszczenia zapisu i implementacji wszystkie zmienne decyzyjne zdefiniowane pod koniec poprzedniego podrozdziału zapisano w jednym wektorze:

$x = [u_{11} \dots u_{1n} u_{21} \dots u_{2n} u_{31} \dots u_{3n}]^T$ - wektor zmiennych decyzyjnych

$Q(x)$ – wartość wskaźnika jakości

Algorytm:

1. Inicjalizacja:

Określ punkt startowy x i wartość $Q(x)$.

Określ parametry $\varepsilon_0, \varepsilon_1, \varepsilon_2, \varepsilon_3$: $\varepsilon_0 > 0, 0 < \varepsilon_1 < \varepsilon_0^2, 0 \leq \varepsilon_2 < 1, \varepsilon_3 > 0$

Podstaw $R:=1$

2. Wylicz gradient wskaźnika jakości $\nabla Q(x)$

Jeśli $\|\nabla Q(x)\| \leq \varepsilon_0 \rightarrow \text{STOP}$

3. Jeśli $R=1$, podstaw $\tilde{W} := I_n$. W przeciwnym razie podstaw:

$$r := \nabla Q(x) - g_s$$

$$s := x - x_s$$

$$\tilde{W} := \tilde{W} + \frac{rr^T}{s^T r} - \frac{\tilde{W} s s^T \tilde{W}}{s^T \tilde{W} s}$$

Wyznacz d z równania: $\tilde{W} \cdot d = -\nabla Q(x)$

4. Jeśli $d^T \nabla Q(x) > -\max(\varepsilon_1, \varepsilon_2 \|\nabla Q(x)\|^2)$, podstaw $R:=1$ i wróć do kroku 3

5. Podstaw:

$$x_s := x$$

$$g_s := \nabla Q(x)$$

6. Wykonaj poszukiwanie w kierunku dz punktu x_s , wyznaczając nowe wartości x i $Q(x)$, takie że $Q(x) \leq Q(x_s)$

7. Jeśli $\|x - x_s\| > \varepsilon_3$, podstaw $R:=0$ i wróć do kroku 2.
 Jeśli $\|x - x_s\| \leq \varepsilon_3$ i $R=1 \rightarrow \text{STOP}$
 Jeśli $\|x - x_s\| \leq \varepsilon_3$ i $R=0$, podstaw $R:=1$ i wróć do kroku 3.

W trakcie działania powyższego algorytmu wykonywane jest poszukiwanie na kierunku d z punktu x_s . Poszukiwanie to wykonywane jest według poniższych kroków.

1. Oznacz:
 Q_0 – wartość wskaźnika jakości w punkcie startowym metody (x_s)
 $z := 1$
2. Obliczenie nowego punktu:
 $x_n := x_s + z \cdot d$
3. Zastosowanie rzutowania na zbiór sterowań dopuszczalny dla każdego elementu wektora x_n .
4. Obliczenie Q_n – wartość wskaźnika jakości w punkcie x_n
5. Jeżeli $Q_n < Q_0 \rightarrow \text{STOP}$
6. Podstawienie $z := 0,5z$
7. Jeżeli $z < 10^{-12} \rightarrow \text{STOP}$. W przeciwnym przypadku powrót do kroku 2

Algorytm BFGS oraz algorytm poszukiwania na kierunku został zaimplementowany w środowisku MATLAB. Algorytm oblicza wielokrotnie wskaźnik jakości oraz jego gradient. Do uzyskania obu tych wartości niezbędne jest wyliczenie trajektorii stanu i trajektorii sprzężonych. Metody pozyskiwania tychże trajektorii opisane w poprzednim podrozdziale również zostały zaimplementowane w środowisku MATLAB.

8.3 Testy algorytmu optymalizacyjnego

W celu zweryfikowania poprawności działania zaimplementowanego algorytmu dokonano analizy zwracanych przez niego wartości sterowania. Dla wartości tych stworzono trajektorie stanu oraz trajektorie sprzężone, na podstawie których z wzorów (8.7 – 8.9) otrzymano trajektorie sterowania optymalnego. Trajektorie te powinny pokrywać się ze sterowaniem otrzymanym z algorytmu.

Dla wszystkich symulacji wartością początkową sterownia było sterowanie neutralne.

Symulacja I

punkt startowy w układzie kartezjańskim: [10, 2, 5]

początkowa wartość wektora stanu: [-10, -2, -5, 0, 0, 0, 0, 0]

czas symulacji: 6 sekund

liczba przedziałów dyskretyzacji sterowania: 10

współczynnik funkcji kary we wskaźniku jakości: $\rho = 1$

wagi sterowań we wskaźniku jakości: $c_Q = \text{diag}\{1 \ 1 \ 1\}$

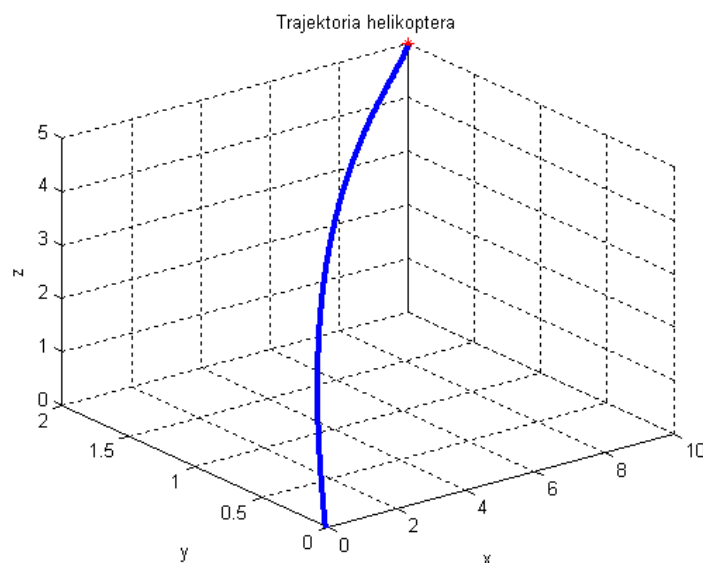
wartość progu algorytmu BFGS: $\varepsilon_0 = 10^{-5}$

Otrzymane wyniki:

wynikowy wskaźnik jakości: $Q = 0.7829$

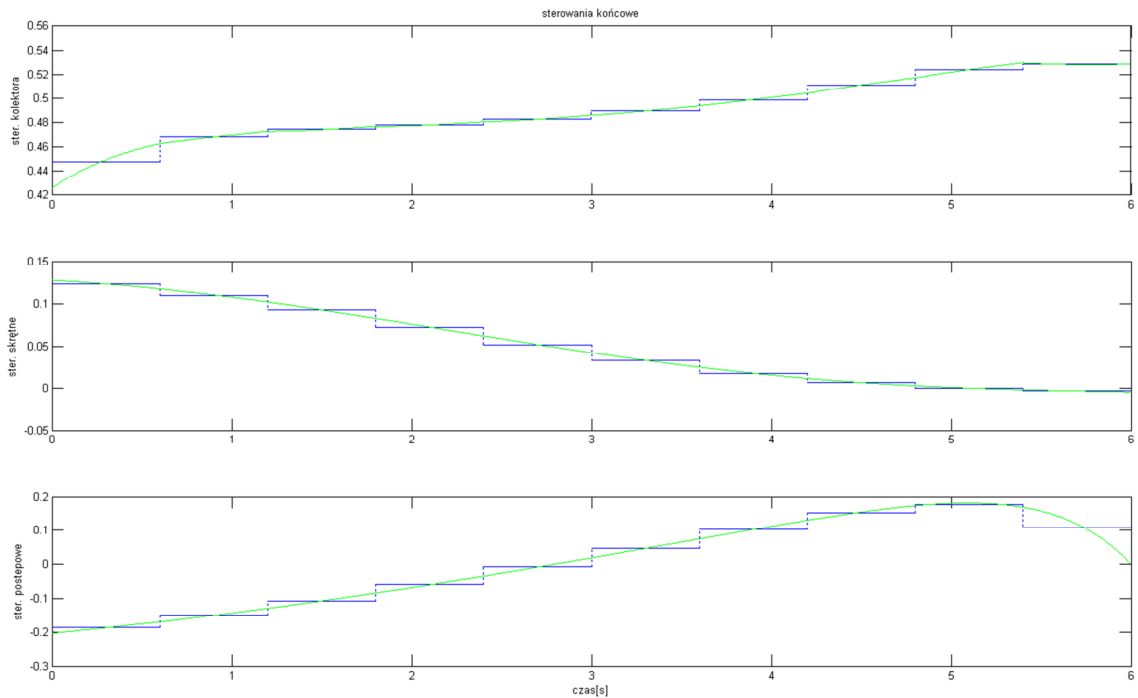
liczba kroków algorytmu BFGS: 59

norma gradientu wskaźnika jakości: $9.5803e-006$



Rysunek 8.1- Trajektorja helikoptera dla symulacji I

Wykreślono sterowanie uzyskane w wyniku działania algorytmu wraz ze sterowaniem optymalnym uzyskanym ze wzorów (8.7 - 8.9).



Rysunek 8.2 - Sterowanie uzyskane z algorytmu wraz ze sterowaniem optymalnym (Symulacja I)

Jak widać sterowanie schodkowe jest zbliżone do optymalnego. Zwiększenie liczby schodków powinno jeszcze bardziej zbliżyć do siebie oba wykresy – przeprowadzono więc kolejną symulację zwiększając liczbę przedziałów do 20.

Symulacja II

punkt startowy w układzie kartezjańskim: [10, 2, 5]

początkowa wartość wektora stanu: [-10, -2, -5, 0, 0, 0, 0, 0]

czas symulacji: 6 sekund

liczba przedziałów dyskretyzacji sterowania: 20

współczynnik funkcji kary we wskaźniku jakości: $\rho = 1$

wagi sterowań we wskaźniku jakości: $c_Q = \text{diag}\{1 \ 1 \ 1\}$

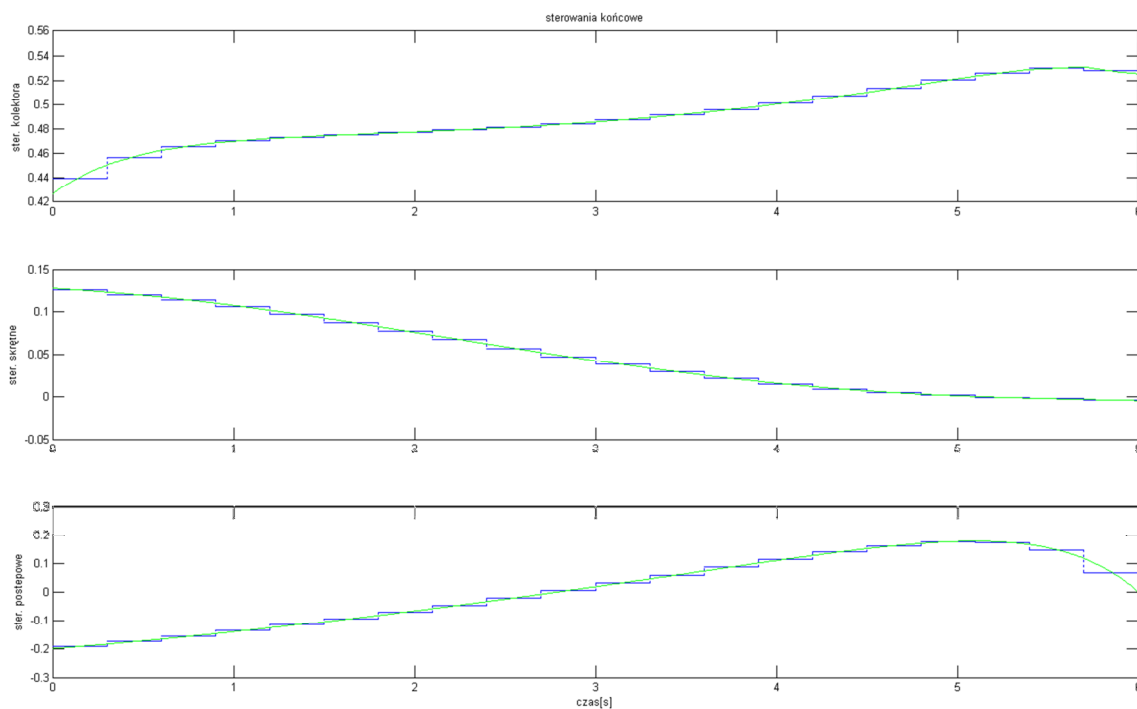
wartość progu algorytmu BFGS: $\varepsilon_0 = 10^{-5}$

Otrzymane wyniki:

wynikowy wskaźnik jakości: $Q = 0.7820$

liczba kroków algorytmu BFGS: 55

norma gradientu wskaźnika jakości: 5.2379e-006



Rysunek 8.3 - Sterowanie uzyskane z algorytmu wraz ze sterowaniem optymalnym (Symulacja II)

Porównując sterowanie otrzymane za pomocą algorytmu BFGS dla obu symulacji można stwierdzić, że algorytm działa poprawnie – sterowanie zwracane przez algorytm zbliżone jest do optymalnego. Ponadto zmniejszając rozmiar „schodka” w optymalizowanym sterowaniu algorytm uzyskuje lepsze wyniki coraz bardziej zbliżając się do ciągłego sterowania optymalnego.

Przeprowadzono kolejną symulację dla innych parametrów w

Symulacja III

punkt startowy w układzie kartezjańskim: [-20, 10, 10]

początkowa wartość wektora stanu: [20, -10, -10, 0, 0, 0, 0, 0]

czas symulacji: 10 sekund

liczba przedziałów dyskretyzacji sterowania: 10

współczynnik funkcji kary we wskaźniku jakości: $\rho = 2$

wagi sterowań we wskaźniku jakości: $c_Q = \text{diag}\{1\ 1\ 1\}$

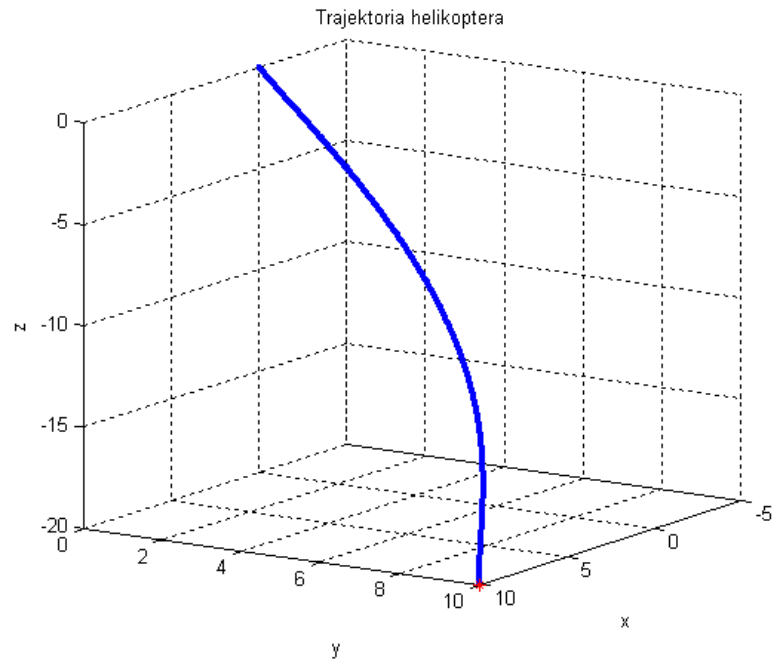
wartość proggu algorytmu BFGS: $\varepsilon_0 = 10^{-5}$

Otrzymane wyniki:

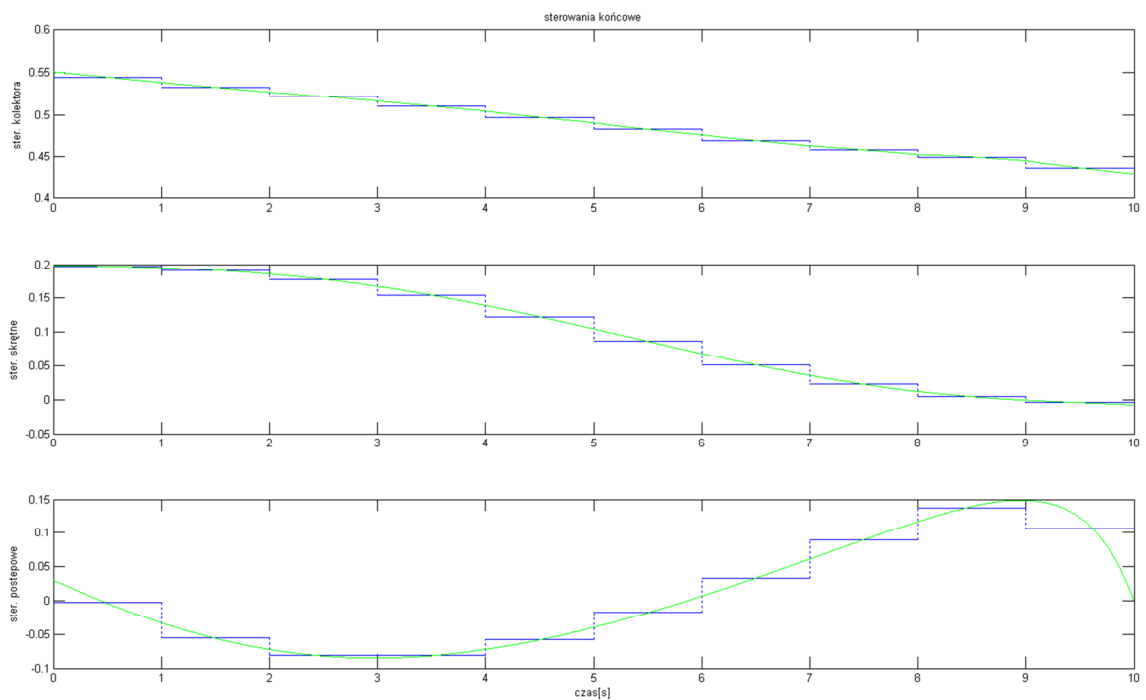
wynikowy wskaźnik jakości: $Q = 1.3166$

liczba kroków algorytmu BFGS: 141

norma gradientu wskaźnika jakości: $9.7276e-006$



Rysunek 8.4 - Trajektoria helikoptera z zaznaczonym punktem startowym (symulacja III)



Rysunek 8.5 - Sterowanie uzyskane z algorytmu wraz ze sterowaniem optymalnym (Symulacja III)

Dla wszystkich trzech symulacji uzyskane sterowanie schodkowe przybliża sterowanie optymalne. Wiedząc, że algorytm działa poprawnie przeprowadzono szereg symulacji, w których badano wpływ poszczególnych parametrów na wyniki zwracane przez algorytm.

Wszystkie symulacje wykonano z punktu startowego helikoptera o współrzędnych [10, -2, 5]. Czas symulacji wynosił $T = 5s$, a próg w metodzie BFGS miał wartość 10^{-5} .

Jako pierwszy analizie został poddany wpływ współczynnika ρ we wskaźniku jakości. Symulacje zostały przeprowadzone dla liczby przedziałów równej 10 oraz przy wagach sterowań $c_Q = \text{diag}\{1 \ 1 \ 1\}$.

Uzyskane wyniki:

ρ	$X(T)^T W X(T)$	Końcowa wartość wskaźnika jakości	Liczba kroków algorytmu
0.002	48.6285	0.5674	9
0.02	3.1897	0.6643	14
0.2	0.0648	0.7046	33
2	7.5571e-004	0.7108	75
20	8.6235e-006	0.7115	141
200	8.9016e-008	0.7116	332
2000	8.9342e-010	0.7116	758

Można zauważyć, że zwiększanie dziesięciokrotnie parametru ρ skutkuje w większości przypadków około stukrotnym zmniejszeniem kwadratu odchyłek stanu końcowego. Zwiększanie wartości ρ powoduje dokładniejsze osiągnięcie zadanego punktu końcowego równocześnie zwiększając liczbę kroków algorytmu, co wiąże się z wydłużeniem czasu jego działania. Pojawia się tu klasyczny problem – poprawienie wyników skutkuje wydłużeniem czasu ich uzyskania. W problemach rzeczywistych

tego typu należy znaleźć kompromis uwarunkowany specyfiką postawionego problemu. Dodatkowo można zauważyć, że dla źle dobranych parametrów (zbyt mała kara za nieosiągnięcie warunków końcowych) nawet poprawnie działający algorytm nie pozwala na rozwiązanie problemu.

Drugim z porównywanych parametrów była liczba przedziałów. Wektor wag sterowań we wskaźniku jakości $c_Q = \text{diag}\{1\ 1\ 1\}$, współczynnik $\rho = 2$.

Uzyskane wyniki:

Liczba przedziałów	Czas działania algorytmu [s]	Końcowa wartość wskaźnika jakości	Liczba kroków algorytmu
5	20.94	0.7160	90
10	12.74	0.7108	75
20	10.23	0.7092	66
50	16.25	0.7087	66
100	11.14	0.7087	60
500	99.86	0.7087	41

Zwiększenie liczby przedziałów odpowiada gładkości sygnału sterującego. Im bardziej sterowanie zbliżone jest do sygnału ciągłego, tym bardziej zbliżone jest do sterowania maksymalizującego hamiltonian. Mniejsza liczba kroków algorytmu świadczy o większej łatwości w znalezieniu sterowania optymalnego. W problemach rzeczywistych liczba przedziałów może być ograniczona przez możliwości sprzętowe.

8.4 Porównanie algorytmów

Stworzony w rozdziale 8 algorytm jest optymalny, jednak stwierdzenie tego faktu nie jest zbyt intuicyjne. W prosty sposób można jednak pokazać, że algorytm ten jest lepszy, niż te zaimplementowane w rozdziale poprzednim.

Do porównania wybrano algorytm mający to samo zadanie co algorytm stworzony w tym rozdziale, czyli lot do punktu. Czas symulacji ustalono tak, aby oba algorytmy były w stanie wykonać powierzone im zadanie.

Symulacja IV

punkt startowy w układzie kartezjańskim: [10, 5, 5]

początkowa wartość wektora stanu: [-10, -5, -5, 0, 0, 0, 0, 0]

czas symulacji: 15 sekund

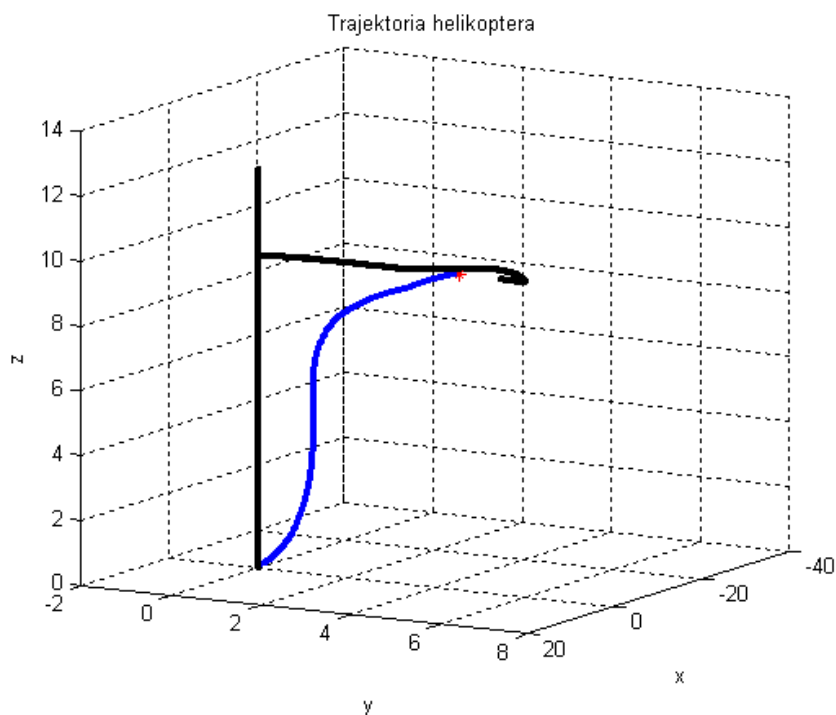
liczba przedziałów dyskretyzacji sterowania: 10

współczynnik funkcji kary we wskaźniku jakości: $\rho = 1$

wagi sterowań we wskaźniku jakości: $c_Q = \text{diag}\{1 \ 1 \ 1\}$

wartość progu algorytmu BFGS: $\varepsilon_0 = 10^{-5}$

Otrzymane trajektorie:



Rysunek 8.6 - Trajektoria helikoptera. Czarna - algorytm lotu do punktu. Niebieska - optymalna

Wskaźnik jakości dla algorytmu lotu do punktu: 14.0447

Wskaźnik jakości trajektorii optymalnej: 2.3227

Wskaźnik jakości jest dużo mniejszy dla sterowania algorytmem optymalnym. Widać także, że dla algorytmu lotu do punktu helikopter bardzo długo się obracał, a dopiero później rozpoczął lot do przodu. Inna sytuacja ma miejsce dla trajektorii optymalnej - ruch postępowy rozpoczyna się mimo nie osiągnięcia zadanego kąta obrotu.

Symulacja V

punkt startowy w układzie kartezjańskim: [-12, 0, 10]

początkowa wartość wektora stanu: [12, 0, -10, 0, 0, 0, 0, 0]

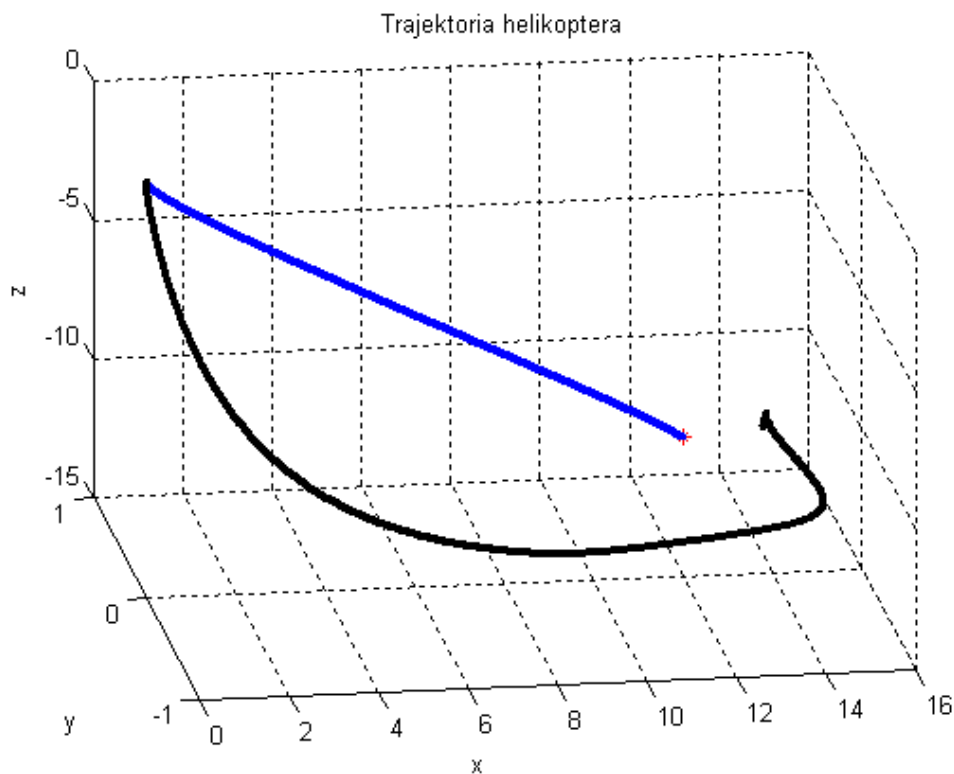
czas symulacji: 10 sekund

liczba przedziałów dyskretyzacji sterowania: 20

współczynnik funkcji kary we wskaźniku jakości: $\rho = 1$

wagi sterowań we wskaźniku jakości: $c_Q = \text{diag}\{1 \ 1 \ 1\}$

wartość progu algorytmu BFGS: $\varepsilon_0 = 10^{-5}$



Rysunek 8.7 - Trajektoria helikoptera. Czarna - algorytm lotu do punktu. Niebieska - optymalna

Wskaźnik jakości dla algorytmu lotu do punktu: 5.3926

Wskaźnik jakości trajektorii optymalnej: 1.2155

Tak samo jak w przypadku symulacji IV lepsze wyniki otrzymano dla sterowania optymalnego. W obu przypadkach trajektoria wynikająca z działania optymalizacji jest łagodniejsza – nie występują przeregulowania, a sterowanie nie przyjmuje wartości z granicy przedziału dopuszczalnego.

Lepsze działanie algorytmu optymalnego wynika z dokładnej wiedzy o zachowaniu analizowanego zagadnienia. Zważywszy na jego skomplikowaną budowę – nieliniowe równania oraz sterowanie drugimi pochodnymi obserwowanych wartości – sterowanie konwencjonalnymi regulatorami jest trudne, a co za tym idzie daje gorsze wyniki.

9. Podsumowanie

Tematem pracy było stworzenie systemu pozwalającego na zdalne sterowanie helikopterem. W ramach pracy należało stworzyć układ wizyjny pozwalający na odczyt niezbędnych informacji oraz układ nadawczy, który po analizie protokołu komunikacji będzie pozwalał na komunikację. Kolejnym krokiem była implementacja wszystkich algorytmów pozwalających na sterowanie helikopterem wraz z algorytmem obliczającym trajektorie optymalną w oparciu o zasadę maksimum Pontriagina.

Pierwszym elementem stworzonym w pracy był system wizyjny. Wybrano system stereowizyjny, dzięki któremu można wyznaczyć współrzędne punktu w trzech wymiarach. Za punkt, który wybrano jako przybliżenie pozycji helikoptera wybrano diodę umieszczoną pomiędzy płozami. Do wykrycia diody niezbędna była odpowiednia obróbka obrazu, której skuteczność została przeanalizowana w zależności od wybranych parametrów. Dla najlepszego zestawienia parametrów uzyskano skuteczność odczytu na poziomie 95%. Oprócz odczytu położenia badano także kąt obrotu śmigłowca wokół osi pionowej. W tym celu zamontowano odpowiednie markery oraz stworzono algorytm obróbki obrazu, który znajdował stworzone markery na obrazie. Algorytm ten poddano podobnej analizie, jak ten wykrywający diodę. Algorytm mimo niezbyt dużej skuteczności na pojedynczych klatkach filmu był w stanie poprawnie śledzić orientację helikoptera w trakcie trwania lotu.

Kolejnym elementem wchodzącym w skład projektowanego systemu był element nadawczy. W ramach tegoż elementu skonstruowano prosty nadajnik i odbiornik podczerwieni. Nadajnik stworzono w celu komunikacji z helikopterem, a odbiornik posłużył w celu identyfikacji protokołu komunikacji na podstawie sygnału z pilota. Obserwując sygnał pochodzący z pilota stwierdzono, że w skład przesyłanego sygnału wchodzi cztery odpowiednio kodowane informacje odpowiadające za siłę ciągu, obrót, ruch postępowy oraz wartość korygującą obrót. W celu wykorzystania do komunikacji biblioteki WinLIRC stworzono odpowiedni plik konfiguracyjny, który obejmował wszystkie możliwe do przesłania komendy. Niestety, przy próbach komunikacji z helikopterem stworzony człon nadawczy nie działał poprawnie – najprawdopodobniej długości impulsów wchodzących w skład protokołu komunikacji są zbyt krótkie, by biblioteka wraz z nadajnikiem była w stanie je przesłać.

Zważywszy na fakt, że nie udało się sterować obiektem rzeczywistym stworzono model matematyczny analizowanego zagadnienia, który wykorzystano w dalszej części pracy.

Stworzono cztery algorytmy, które pozwalały na sterowanie modelem helikoptera. Pierwszym z nich był algorytm sterowania neutralnego, który miał za zadanie utrzymywać helikopter na zadanej wysokości. Algorytm ten mógł też regulować tę wysokość podczas bardziej skomplikowanego lotu. Drugi z algorytmów polegał na stabilizacji w punkcie. Algorytm nadzorował położenie helikoptera i przy zbyt dużej odległości od punktu zadanego następował obrót w kierunku punktu docelowego i lot w jego kierunku. Trzecim i zarazem ostatnim z konwencjonalnych algorytmów była procedura lotu do punktu. Jest to pewna wariacja algorytmu stabilizacji. Helikopter obracał się w kierunku zadanego punktu i wykonywał lot postępowy wraz z regulacją wysokości do wartości zadanej. Wszystkie z wymienionych algorytmów nie działają jednak idealnie – jest to spowodowane dużym poziomem skomplikowania modelu oraz faktem, że sterowania oddziałują nie bezpośrednio na wartości obserwowane, a na ich drugie pochodne.

Czwarty z algorytmów wykorzystywał zasadę maksimum Pontriagina. Stworzono wskaźnik jakości który odzwierciedlał energię zużytą przez śmigłowiec na lot wraz z karą za niespełnienie warunków końcowych. Za warunek końcowy postawiono dolecenie helikoptera do punktu zerowego wraz z wyhamowaniem wszystkich prędkości do poziomu zera. Poszukiwań sterowania optymalnego dokonano z wykorzystaniem algorytmu Broydena – Fletchera – Goldfarba – Shanno. Analizując wyniki zwrócono uwagę na zależność między otrzymanymi wynikami, a parametrami wchodzącymi w skład algorytmu i wskaźnika jakości. Przy analizie otrzymano klasyczny problem występujący w optymalizacji – zmiana niektórych parametrów powoduje wzrost dokładności kosztem czasu trwania algorytmu. Przy ich doborze należy znaleźć kompromis zależny od celu oraz mocy obliczeniowej.

Porównując wyniki otrzymane z algorytmu optymalnego z tymi otrzymanymi z algorytmu konwencjonalnego dokonano zadowalających obserwacji. Dla wszystkich przeprowadzonych symulacji wskaźnik jakości miał dużo mniejszą wartość dla trajektorii lotu otrzymanej z algorytmu BFGS.

Podsumowując system stworzony w pracy nie jest kompletny, aczkolwiek poszczególne elementy, z wyjątkiem elementu nadawczego, zostały przetestowane, a ich działanie jest zadawalające. Sam nadajnik podczerwieni też działa poprawnie, jednak jego dynamika jest niewystarczająca dla wybranego zagadnienia.

Praca zapewnia wiele możliwości jej rozwoju, w tym między innymi:

- Wykorzystanie kamer lepszej jakości pozwoliłoby na zwiększenie użytej rozdzielczości, a co za tym idzie dokładności pomiarów. Ponadto lepsza rozdzielczość pozwala na odsunięcie kamer, co zwiększa przestrzeń, w której może latać helikopter.
- W polepszeniu wyników rozpoznawania orientacji mogą pomóc lepiej widoczne markery, które w czasie obróbki byłyby dokładniej rozpoznawane, bądź inny ich kształt lub umiejscowienie.
- Ważnym elementem byłaby konstrukcja poprawnie działającego elementu nadawczego. W tym celu można stworzyć osobny system z mikroprocesorem, do którego komputer przesyłałby tylko komendy. Mikroprocesor zajmowałby się tylko odpowiednim załączaniem/wyłączaniem diody.
- Model matematyczny również można ulepszyć – model wykazuje mniejszą samohamowność niż obiekt rzeczywisty. Ponadto model nie uwzględnia poziomu zużycia baterii.
- Algorytmy zaimplementowane w ramach pracy nie obejmują wszystkich możliwości użytego helikoptera. W trakcie rozwoju pracy można by stworzyć algorytmy pozwalające na ruch po okręgu, czy innej zadanej trajektorii.

10. Bibliografia

- [1] P. Rotter. *Transformacje kinematyczne i wizyjne w robotyce – wykłady*. AGH
- [2] G. Gilewska. *Przetwarzanie i rozpoznawanie obrazów – wykłady*. Politechnika Białostocka
- [3] R. Tadeusiewicz, P. Kohorda. *Komputerowa analiza i przetwarzanie obrazów*. AGH
- [4] B. Cyganek. *Komputerowe przetwarzanie obrazów trójwymiarowych*. Akademicka Oficyna Wydawnicza EXIT, Warszawa 2002
- [5] M. Dajda, G. Łojek. *Zastosowanie stereowizji do odtwarzania trajektorii obiektów w przestrzeni 3D*. AGH
- [6] *GaAs Infrared Emitter*. OSRAM Opto Semiconductors GmbH & Co. OHG
- [7] *Photo Modules for PCM Remote Control Systems*. Vishay Telefunken
- [8] <http://winlirc.sourceforge.net/>
- [9] R. Cunha, C. Silvestre, A. Pascoal, I. Kaminer. *Path Following Control for an Autonomous Helicopter*. Lisbon, Portugal 2002.
- [10] A. Korytowski. *Wykłady do przedmiotu Optymalizacja w Systemach Sterowania na kierunku Automatyka i Robotyka w semestrze pierwszym na studiach drugiego stopnia*. AGH.

11. Załączniki

- [1] Płyta CD-ROM – tekst pracy, pliki źródłowe, filmy i biblioteki

Dodatek A – Spis materiałów zawartych na płycie

Poniżej zestawiono wszystkie pliki zawarte na płycie CD wraz z ścieżkami do folderów, w których się znajdują oraz krótkim opisem.

- CD:\Tekst – Tekst pracy w postaci pliku Word oraz PDF.
- CD:\Biblioteki\WinLIRC – Biblioteka WinLIRC wraz ze wszystkim wchodzącymi w jej skład programami oraz plik konfiguracyjny.
- CD:\Biblioteki\WinLIRC65 – Starsza wersja biblioteki.
- CD:\Filmy\Oryginalne – Filmy z nagraniem lotu helikoptera. Podzielone na katalogi tak, aby filmy nagrywane bez zmian położenia kamer znajdowały się w jednym katalogu. W Folderach zawarto także obrazy zerowe, orientacje zerowe helikoptera oraz obrazy ze wzorcem kalibracyjnym.
- CD:\Filmy\Obrobione\Dioda – Filmy po obróbce w celu wykrycia diody.
- CD:\Filmy\Obrobione\Orientacja – Filmy po obróbce z wykrytym prostokątem.
- CD:\Programy\Dioda bez odejmowania – Program zaimplementowany w języku MATLAB/Simulink pozwalający na obróbkę w celu znalezienia diody i zapisanie filmu. Dzięki programowi stworzono filmy omówione w 4.1.
- CD:\Programy\Dioda z odejmowaniem – Program zaimplementowany w języku MATLAB/Simulink pozwalający na obróbkę w celu znalezienia diody i zapisanie filmu. Dzięki programowi stworzono filmy omówione w 4.2.
- CD:\Programy\Stereowizja – Program, który po obróbce w celu znalezienia diody wyświetla trajektorię, po której poruszał się śmigłowiec. Wykorzystany do testu stereowizji omówionego w 4.3
- CD:\Programy\Wykrycie prostokata – Program zaimplementowany w języku MATLAB pozwalający na obróbkę w celu wykrycia prostokąta i zapisanie filmu. Dzięki programowi stworzono filmy omówione w 4.4.
- CD:\Programy\Nagrywanie – Program pozwalający na jednoczesne nagrywanie filmu z obu kamer. Dostępne wersje MATLAB oraz java.
- CD:\Programy\Konfiguracja – Program tworzący plik konfiguracyjny.
- CD:\Programy\Optymalizacja – Program w środowisku MATLAB pozwalający na obliczenie sterowania optymalnego dzięki algorytmowi BFGS (rozdział 8).
- CD:\Programy\Algorytmy – Model helikoptera wraz z algorytmami sterującymi (rozdział 7).

W każdym z powyższych folderów znajdują się plik tekstowy z listą plików i krótką instrukcją