Bulgarian Academy of Sciences. Space Research and Technology Institute.
Aerospace Research in Bulgaria. 35, 2023, Sofia

DOI: https://doi.org/10.3897/arb.v35.e12

RETRACTABLE UNDERCARRIAGE DESIGN FOR A QUADCOPTER

Konstantin Metodiev

Space Research and Technology Institute – Bulgarian Academy of Sciences
e-mail: komet@space.bas.bg

Keywords: *Undercarriage, Quadcopter, Autodesk inventor, Four bar linkage*

Abstract

In the current paper, design of a foldable undercarriage for quadcopter is suggested and investigated. The deploying/retracting mechanism comprises a four-bar linkage. The mechanism lowermost configuration forms a kinematic lock. A solution to complex variable equations has been worked out to put together the mechanism geometry. In addition, a 3D CAD model has been developed to visualize the design better.

Introduction

Retractable landing gear design is primarily employed to reduce drag of the aircraft. It is somewhat unfortunate that the retractable landing gear may alter longitudinal balance of the aircraft. In addition, the actuating mechanism adds extra weight. Despite a few shortcomings, the retractable landing gear is indispensable part of a high-speed airplane.

In case of a quadcopter, the undercarriage design is specific. It must meet following expectations:

- not to obstruct the payload (camera) field of view
- not to collapse whenever the servo motor is switched off, i.e. a kinematic lock is required
- to protect the payload (gimbal camera)
- to meet stress and strain requirements: yield, fatigue, deformation, etc.
- to be lightweight

The project goal is to devise a two-dimensional mechanism for undercarriage deploying/retracting that meets the aforementioned design requirements. Source code has been developed in C and build by TDM-GCC [1] to work out a solution to kinematic design equations with regard to a complex variable. In addition, it was rather necessary to develop a 3D CAD model to fulfil the project objectives better. In the current research, a trial version of Autodesk Inventor [2] has been used to investigate the construction thoroughly.

Method

Design milestones are described below. The design comprises two stages: developing a source code in C to fulfil kinematic analysis (see Appendix section) and a CAD model to test the solution.

Analysis

Given prescribed rotations, it is desirable for the four-bar linkage to obey the Grashof's law:

$$s + l \leq a + b \tag{1}$$

where variables stand for following: s – the shortest link, l – the longest link, a, b – remaining links. If the shortest link is adjacent to the fixed link, the mechanism is called "a crank rocker four-bar linkage." However, in the current project, the shortest link is unable to perform a full revolution. It is not necessary either.

The four-bar linkage analysis follows algorithm described in textbook [3]. Standard dyad form of the governing equations has been employed according to following expression and Fig. 1:

$$\mathbf{W}\left(e^{\mathbf{i}\varphi_j}-1\right)+\mathbf{Z}\left(e^{\mathbf{i}\gamma_j}-1\right)=\boldsymbol{\delta}_j=\mathbf{r}_j-\mathbf{r}_{j-1}=\mathbf{X}\left(e^{\mathbf{i}\psi_j}-1\right),\quad j=1,2,3 \tag{2}$$

In eq. (2) angles φ_j, γ_j, j = 1, 2, 3 are prescribed in advance. Values are assigned to displacement vector $\boldsymbol{\delta}_j$ by trial and error. Vectors **W**, **Z**, and **X** are complex.

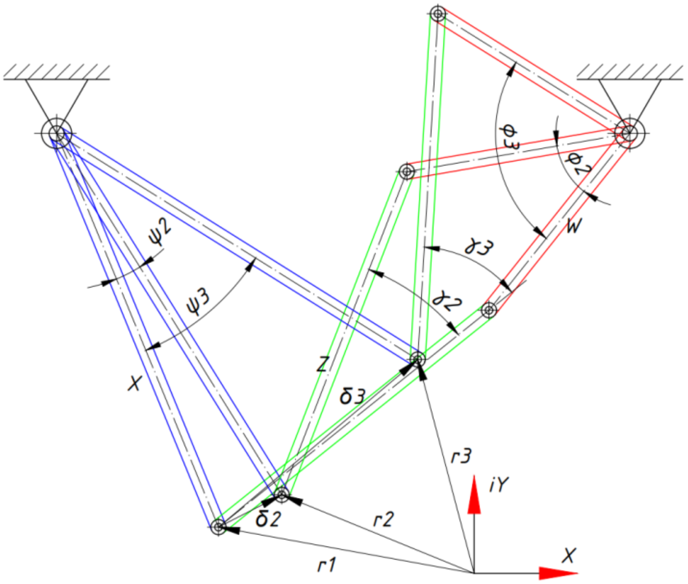

Fig. 1. Three initial configurations of the four-bar linkage

Considering eq. (2), it is straightforward to work out vector exp(**i**θ) modulus by means of Euler's identity (**i** = √–1, θ is an arbitrary angle):

(3) $$\left|e^{\mathbf{i}\theta}\right| = \left|\cos(\theta) + \mathbf{i}\sin(\theta)\right| = \sqrt{\cos^2(\theta) + \sin^2(\theta)} = 1$$

Therefore, having been multiplied by exp(**i**θ), a vector rotates through angle θ.

Eq. (2) could be written down taking into account each initial position.

(4) $$\begin{aligned}\mathbf{W}\left(e^{\mathbf{i}\varphi_2} - 1\right) + \mathbf{Z}\left(e^{\mathbf{i}\gamma_2} - 1\right) &= \boldsymbol{\delta}_2 \\ \mathbf{W}\left(e^{\mathbf{i}\varphi_3} - 1\right) + \mathbf{Z}\left(e^{\mathbf{i}\gamma_3} - 1\right) &= \boldsymbol{\delta}_3\end{aligned}$$

This is a system of equations linear in the complex unknown vectors **W** and **Z**. It might be solved by Cramer's rule. A source code is available in Appendix section.

(5) $$\mathbf{W} = \frac{\begin{vmatrix}\boldsymbol{\delta}_2 & e^{i\gamma_2} - 1 \\ \boldsymbol{\delta}_3 & e^{i\gamma_3} - 1\end{vmatrix}}{\begin{vmatrix}e^{i\varphi_2} - 1 & e^{i\gamma_2} - 1 \\ e^{i\varphi_3} - 1 & e^{i\gamma_3} - 1\end{vmatrix}}; \quad \mathbf{Z} = \frac{\begin{vmatrix}e^{i\varphi_2} - 1 & \boldsymbol{\delta}_2 \\ e^{i\varphi_3} - 1 & \boldsymbol{\delta}_3\end{vmatrix}}{\begin{vmatrix}e^{i\varphi_2} - 1 & e^{i\gamma_2} - 1 \\ e^{i\varphi_3} - 1 & e^{i\gamma_3} - 1\end{vmatrix}}$$

Assembly

The mechanism outline is shown in Fig. 2. A four-bar linkage is clearly visible. It is made up of two cranks (dark blue and dark red), a rocker (green), and a support (yellow) at which the entire mechanism is suspended. Input link is the dark red one which is actuated by a servo motor. It is important to note that both input crank and rocker are collinear on purpose at the linkage dead center. This is a configuration at which the transmission ratio is minimum. The kinematic lock imposes a constraint on the entire mechanism in deployed configuration. It is formed by a zero-force member (a fork) right beneath the middle axis, Fig. 2 - left.

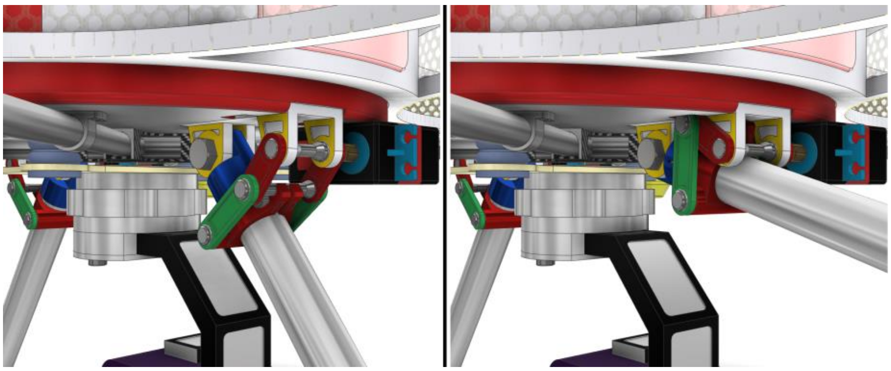

Fig. 2. The undercarriage fulcrum: deployed (left) and retracted leg

Results

In Fig. 3, both deployed and retracted mechanism configurations are depicted. Although the retracted rods are not quite horizontal, it is evident that the gimbal motion is not restricted, neither is the camera field of view.

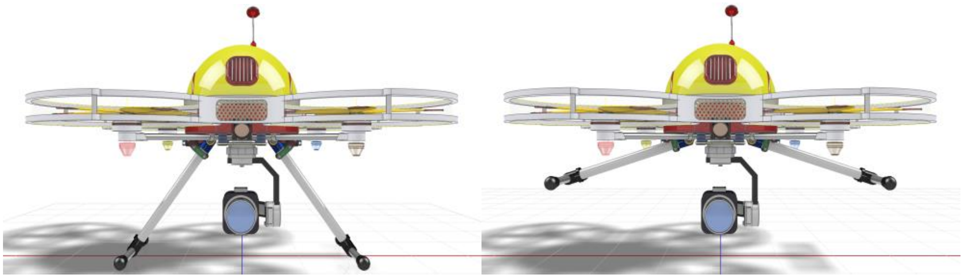

Fig. 3. Deployed (left) and retracted configuration of the undercarriage

Initial values, according to notation adopted in Fig. 1, follow. Rotation counterclockwise is positive.

$\mathbf{r}_1 = -16.949 + \mathbf{i}*3.068$; $\mathbf{r}_2 = -12.753 + \mathbf{i}*5.215$; $\boldsymbol{\delta}_2 = -\mathbf{r}_1 + \mathbf{r}_2$

$\mathbf{r}_3 = -3.739 + \mathbf{i}*14.196$; $\boldsymbol{\delta}_3 = -\mathbf{r}_1 + \mathbf{r}_3$

$\varphi_2 = -41.75$ deg; $\varphi_3 = -83.5$ deg; $\psi_2 = 30.101$ deg; $\psi_3 = 47.893$ deg

After building and executing the source in Appendix section, following values were obtained (console output):

```
W = -9.34-11.74i
Z = -17.93-14.40i
|W| = 15.00     |Z| = 23.00

Process returned 0 (0x0) execution time: 0.018 s
Press any key to continue.
```

Length of both input crank W and rocker Z are exact. There is a full agreement between analytical results and blueprint.

Conclusion

Deployed linkage configuration is considered stable enough. A kinematic lock (a.k.a. downlock) prevents the deployed mechanism from collapsing. Hence, the actuating servo motor might be switched off safely.

Further project development, according to Fig. 4, is to carry out stress and strain analyses.

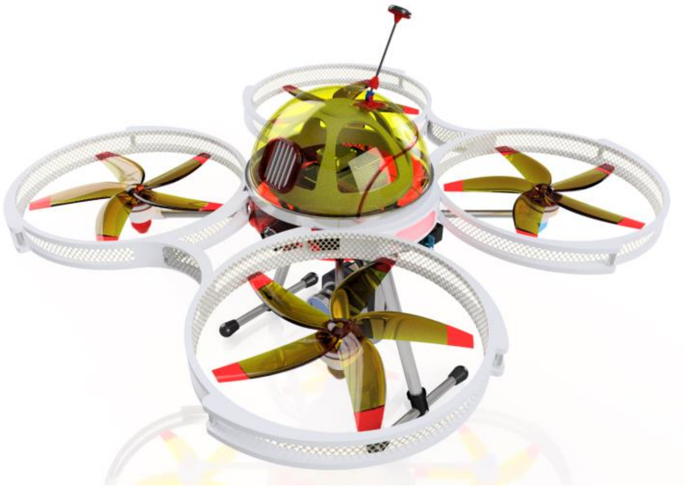

Fig. 4. Perspective projection of the quadcopter [4]

References

1. https://jmeubank.github.io/tdm-gcc/articles/2021-05/10.3.0-release
last visit on 29th of June, 2022
2. https://www.autodesk.com/products/inventor/overview
last visit on 29th of June, 2022
3. Erdman, A. G., G. N. Sandor, S. Kota, Mechanism Design, Analysis and Synthesis, Vol. 1, p. 573, Fourth Edition, Prentice Hall, ISBN 0-13-040872-7
4. https://grabcad.com/library/quadcopter-234
last visit on 29th of June, 2022

ПРОЕКТ НА ПРИБИРАЕМА СТОЙКА ЗА КВАДРОКОПТЕР

К. Методиев

Резюме

В настоящата статия е предложен и изследван проект на прибираема стойка за квадрокоптер. Механизмът се състои от шарнирен четиризвенник. Крайната конфигурация на механизма образува кинематична ключалка. Механизмът бе синтезиран посредством решаване на уравнения с комплексна променлива. В допълнение бе разработен 3D CAD модел за по-добра визуализация на механизма.

Appendix. Source code in C for working out a solution to eq. (5), Fig. 1

```c
#include <stdio.h>
#include <complex.h>
#include <stdlib.h>
#define PI 3.14159265358979323846264338327 95

typedef double complex cplx;

cplx* Cramer(double *a, cplx d2, cplx d3); // forward declaration

int main() {

    cplx r1 = -16.949 + 3.068*I;
    cplx r2 = -12.753 + 5.215*I;
    cplx d2 = -r1 + r2;

    cplx r3 = -3.739 + 14.196*I;
    cplx d3 = -r1 + r3;

    double a[4] = {-41.75, -83.5, 30.101, 47.893}; // phi, psi
    double *pa = &a[0];

    cplx *ans = Cramer(pa, d2, d3);

    printf("W = %.2f%+.2fi\n", creal(ans[0]), cimag(ans[0]));
    printf("Z = %.2f%+.2fi\n", creal(ans[1]), cimag(ans[1]));
    printf("|W| = %.2f\t|Z| = %.2f\n", cabs(ans[0]), cabs(ans[1]));

    free(ans);

    return 0;
} // main

cplx* Cramer(double *a, cplx d2, cplx d3) {

    for (int i = 0; i < 4; i++) a[i] *= PI / 180.;

    cplx W = (d2 * (cexp(a[3]*I) - 1.) - d3 * (cexp(a[2]*I) - 1.)) /
    ((cexp(a[0]*I) - 1.) * (cexp(a[3]*I) - 1.) -
    (cexp(a[2]*I) - 1.) * (cexp(a[1]*I) - 1.));

    cplx Z = ((cexp(a[0]*I) - 1.) * d3 - (cexp(a[1]*I) - 1.) * d2) /
    ((cexp(a[0]*I) - 1.) * (cexp(a[3]*I) - 1.) -
    (cexp(a[1]*I) - 1.) * (cexp(a[2]*I) - 1.));

    cplx* foo = (cplx*)calloc(2, sizeof(cplx));
    foo[0] = W;
    foo[1] = Z;

    return foo;
} // Cramer
```