Anleitung zum Parametrisieren von Flugmodellen für AeroFly5

Von der 3D Zeichnung zum perfekt eingestellten Modell

Aerofly5 von IPACS ist ein sehr leistungsfähiger Modellflugsimulator für PC und MAC. Über Parameterdateien kann dem Simulator die Physik, die Aerodynamik, die Optik und der Sound eines Flugmodells mitgeteilt werden. Diese Anleitung soll das notwendige Hintergrundwissen zum Verständnis und zur Erstellung dieser Parameter vermitteln.

Autor: Andreas Preissler Stand: 17.11.2010

INHALT

EINLEITUNG

Zielgruppe

Diese Anleitung ist für Entwickler gedacht, die für den Modellflugsimulator Aerofly5 von IPACS im passenden 3D Format vorliegende Modelle mit den nötigen und gewünschten Eigenschaften versehen wollen.

Voraussetzungen

Grundsätzliches Verständnis für die physikalischen und aerodynamischen Eigenschaften eines Flugmodells werden erwartet. Zum Teil wird der im Modellbau übliche Sprachgebrauch ohne weitere Erläuterungen verwendet. Es ist daher von Vorteil, mit dieser Materie vertraut zu sein.
Da die meisten Begriffe innerhalb der Parameterdateien in englischer Sprache gehalten sind, ist ein Grundwortschatz Bedingung. Übersetzungen werden in dieser Anleitung nur vereinzelt geliefert oder verwendet.

Der grundlegende Umgang mit dem der Zeichnung zu Grunde liegenden 3D Zeichenprogramm wird ebenso vorausgesetzt.

Über diese Anleitung

Das Aneignen von neuem Wissen geht meist am einfachsten über konkrete Beispiele. Diese Anleitung nimmt das Modell PT40 aus dem AeroFly5 als Demonstrationsobjekt. Aerofly5 gibt es als Versionen für PC und MAC. Diese Anleitung bezieht sich immer auf die PC Variante. Angaben von zum Beispiel Verzeichnisstrukturen, sind also ggf. entsprechend umzusetzen.

Zur besseren Übersicht werden Textteile und Auszüge aus den Parameterdateien in nichtproportionaler Schrift dargestellt. Beispiel:

```
<[float64][K][1837.695190]>   // Kraftkonstante
```

Die Anleitung beginnt mit einem erklärenden Teil, in dem Grundlagen vermittelt werden. Im zweiten Teil geht es dann um die konkrete Vorgehensweise bei der Parametrisierung. Da im ersten Teil erläutert wird, welche „Denkweise“ der Simulator verwendet, ist dieses Kapitel sehr wichtig zum Verständnis und in Folge zur selbstständigen Erstellung von Parameterdateien. Der zweite Teil soll als roter Faden bei der Parametrisierung dienen. Hier sind auch Tipps und Kniffe sowie weitere Hintergrundinfos zu finden.

Das Handwerkszeug (Tools)

Als erstes ist hier natürlich der Flugsimulator selbst zu nennen. Ab der Version 5.5 sind dann auch allgemein diverse Hilfsmittel zugänglich, ohne die das Parametrisieren eher umständlich ist. Die Versionsnummer findet man unter dem Menüpunkt „Hilfe“ im Untermenü „Über AeroFly...“

Alle Dateien, die Parameter für ein Modell enthalten, sind reine Textdateien. Zum Erstellen oder Bearbeiten dieser Dateien reicht also ein einfacher Text-Editor. Notepad, welches bei Windows enthalten ist, kann hierfür verwendet werden. Je höherwertig allerdings der Editor ist, umso einfacher kann man sich die Arbeit machen. Editoren mit Syntax Highlighting beispielsweise können so aufgesetzt werden, dass feste Namen oder Bezeichnungen in bestimmten Farben angezeigt werden. So sieht man gleich bei der Eingabe, ob ein Wort richtig geschrieben wurde und folglich vom Simulator erkannt werden kann. Textbausteine, mit denen man ganze Objekte als Vorlagen einfügen kann, sind ein weiterer Vorteil von höherwertigen Editoren. Notepad++ als Vertreter kostenloser Software sei hier genannt.

```
<[string8][object][rigidbody]
    <[string8][Name][Fuselage]>
    <[float64][Mass][1.600000]>
    <[tmvector3d][InertiaLength][1.150000 0.100000 0.150000]>
    <[tmmatrix3d][Inertia][0.004333 0.000000 0.000000 0.000000 0.179333 0.000000 0.000000 0.000000 0.177667]>
    <[tmvector3d][RO][-0.034669 0.000000 -0.065312]>
    <[tmmatrix3d][BO][1.000000 0.000000 0.000000 0.000000 1.000000 0.000000 0.000000 0.000000 1.000000]>
    <[tmvector3d][CO][0.000000 0.000000 0.000000]>
>
```

Das rigidbody Objekt „Fuselage“ aus der PT40 TMD Datei im Notepad++ mit Syntax Highlighting

Da beim Erstellen der Parameter auch mal was schief gehen kann, ist es sinnvoll, sich alte Dateistände auf zu bewahren. So kann man gegebenenfalls leicht auf eine Version zurückgreifen, bei der noch alles funktioniert hat. Version Management Systeme, also Systeme die Versionen verwalten, bieten diese Möglichkeit. Auch hier gibt es im Bereich der freien Software gute Tools, wie zum Beispiel SVN. SVN alleine ist kommandoorientiert und daher eher etwas für Insider, mit der Shell-Erweiterung TortoiseSVN wird es jedoch zu einem für unsere Zwecke gut einzusetzenden Werkzeug.

Hat man mehrere Versionen eines Modells, so braucht man noch eine Möglichkeit, diese Versionen leicht zu vergleichen. Auch dazu gibt es eine Menge Hilfsmittel am freien Software Markt. Wer sich für Notepad++ als Editor entschieden hat, der hat das Vergleichswerkzeug bereits mit installiert.

Zur Erstellung der 3D Zeichnungen eignen sich die Programme 3DS MAX 9, 3DS MAX 2010, Cinema 4D, AC3D und Metasequoia. Für alle diese Programme gibt es Konverter, welche die 3D Zeichnungsdaten in die von AeroFly5 verwendeten TGC und TGB Formate wandeln. Wir konzentrieren uns hier auf die Programme Metasequoia und AC3D, da diese als Freeware oder günstige Shareware zu bekommen sind. Die entsprechenden Konverter sind dann af5-mqo-to-tgc.exe und af5-ac3d-to-tgc.p. Mit den Konvertern werden zwar die Zeichnungsdaten, aber nicht die Texturen und Bitmaps konvertiert. Hierzu gibt es ein eigenes Tool namens af5-makemaps.exe. Es wandelt Bitmap Dateien in AeroFly5 TTX Dateien. Diese Konverter und Tools gibt es auf der Aerofly Homepage unter www.aerofly.com zum Download.

Last but not least wird natürlich ein leistungsfähiger Rechner mit entsprechender Graphikkarte benötigt. Generell ist jeder Rechner als Entwicklerrechner geeignet, auf dem Areofly5 läuft.

Anforderungen an die 3D Zeichnung

Mit der Parametrisierung kann erst begonnen werden, wenn alle Daten im AeroFly5 Format vorliegen. Dass es dazu Konverter gibt, wissen wir bereits; es gibt jedoch ein paar Dinge, die Voraussetzung für die erfolgreiche Konvertierung und Anwendung in AeroFly5 sind.

1. Geschlossenes Mesh
Alle Meshes müssen in sich geschlossene Objekte sein. Was unter geschlossen zu verstehen ist, verdeutlicht am besten die Vorstellung, man würde den Körper mit Wasser füllen. Sobald kein Wasser mehr aus dem Körper ausläuft, ist die Bedingung erfüllt. Ausnahme sind ineinander laufende Flächen (faces). Diese würden zwar die Forderung nach geschlossenen Meshes erfüllen, werden aber vom Konverter nicht akzeptiert. Zu beachten ist auch, dass pro Objekt nur eine Textur verwendet werden darf.

2. Aufteilung
Teile, die sich später am Modell bewegen sollen, müssen auch als Einzelteile gezeichnet werden. Rumpf, Flügel, Ruderflächen, Fahrwerksteile, aber auch Servohörner und Gestänge (Linkages).

3. Namensgebung
Objektnamen müssen logisch benannt werden. Dieser Punkt hat nicht direkten Einfluss auf die Konvertierung und Funktion im Simulator, er ist jedoch für eine geordnete und nachvollziehbare Vorgehensweise sehr wichtig. Sinnvolle Namen sind beispielweise: Fuselage / LeftWing / LeftAileron / LeftFlap / LeftFlap1 / LeftStabilizer / LeftElevator / Stabilizer / Rudder / LeftGear / LeftWheel / Propeller / Blade1 / Blade2 / Cockpit / Canopy / Glass / Rotor / TailRotor / TailBlade1 / TailBlade2 / Antenna / Engine / ....

4. Achsen
Die Achsen oder Drehpunkte (in den meisten Programmen „Pivot“ genannt), sollten so eingerichtet sein, dass man über sie die einzelnen Teile so bewegen kann wie im Original. Der Achsenpunkt des Rumpfes muss allerdings im Schwerpunkt des Flugzeuges liegen.

5. Schwerpunkt
Der Schwerpunkt des Flugzeuges muss in der Mitte des virtuellen Raumes liegen, also auf den Koordinaten X = 0.0, Y = 0.0, Z = 0.0.

6. Texturen
Texturen müssen als 24-bit BMP Dateien vorliegen. Die Texturen werden gegliedert in:

*_color.bmp Für die Bemalung des Modells mit Beschriftung, Armaturen, Farbeffekten etc.

*_alpha.bmp Für die Steuerung von Transparenz, zum Beispiel für Glas, Gitter usw. Es ist darauf zu achten, dass für Objekte mit Transparenz eine separate Textur mit

Alpha Kanal angelegt wird, da es sonst auch bei den Objekten, die keine Transparenz haben sollen, zu Darstellungsfehlern führen kann.

*_bump.bmp Für die Graustufen-Map, mit der man die Erhöhung von Flächen steuern kann, auch Heightmap oder Bumpmap genannt. Damit lassen sich Details wie Schrauben und Nieten polygonsparend darstellen.

*_gloss.bmp Für die Steuerung von Glanz auf Oberflächen. Mittels Grauabstufungen kann man die Intensität des Glanzes steuern, wobei weiß am höchsten glänzt und schwarz gar nicht glänzt. Grauwerte oberhalb der Mitte erzeugen metallischen Glanz, wobei weiß reinen Chromglanz darstellt.

7. Preview Datei

Für die Modellübersicht des Simulators wird eine Vorschaudatei benötigt. Diese muss bestimmte Kriterien erfüllen. Sie bekommt den Namen preview.bmp und wird als reine drei Komponenten BMP mit einer Auflösung 256 mal 256 Punkten angelegt. Der Hintergrund muss weiß sein und das Modell in einem Winkel von 45 Grad von links vorne nach rechts hinten und 30 Grad von oben abgebildet sein.

Anwendung der Konverter

Grundsätzlich wird für die Ausführung aller Konverter eine IPACS Lizenzdatei benötigt. Diese erhält man über den Link http://www.aerofly.com/afpd/downloads.html. Dort kann man sich in der Member Area als Mitglied ein Konto einrichten und dann im Bereich „Tools“ die Personal License Datei herunterladen. Diese muss dann je nach 3D Anwendung ins Verzeichnis des Plug-ins oder der EXE Datei kopiert werden.

Für das Programm AC3D gibt es ein Plug-In namens af5-ac3d-to-tgc.p. Dieses muss in das Plug-in-Verzeichnis von AC3D kopiert werden. Unter „Datei Exportieren“ kann dann mit der Option „AeroFly5 TGC files...“ die TGC Datei erzeugt werden.

Für Metasequoia ist der Konverter ein Kommandozeilentool mit dem Namen af5-mqo-to-tgc.exe. Zur Konvertierung von Metasequoia in TGC ist also erst einmal eine Kommandozeilen Eingabe zu erzeugen. Dies geht einfach über den Windows Start Knopf und dort auf die Auswahl „Ausführen...“. Das zu öffnende Programm heißt „cmd“. Sobald mit Ok quittiert wurde, öffnet sich ein Fenster, in dem Kommandozeilen eingegeben werden können. Hier ist mit Hilfe des Kommandos „cd <verzeichnisname>“ das Verzeichnis zu wählen, in dem die Metasequoia Dateien und der Konverter liegen. Mit dem Kommando Af5-mqo-to-tgc.exe <modellname> (Modellname ohne mqo extension!) wird dann der Konverter gestartet und die TGC Datei erzeugt. Optional kann noch ein Maßstab (scalefactor) angeben werden, falls das Modell in einem anderen Maßstab modelliert wurde. Das Kommando lautet dann:
Af5-mqo-to-tgc.exe <modellname_> scalefactor

Eine eigene Rolle spielen die Texturen und die Steuerdateien für Glanz, Flächenerhöhungen und Transparenz sowie die Preview Datei. Als Konverter dient das Programm af5-makemaps.exe. Alle Dateien, inklusive die des Konverters, müssen in einem Verzeichnis stehen. Wir dann das Programm af5-makemaps.exe gestartet, so sucht es nach den bekannten BMP Dateien und wandelt alle in TTX Dateien um. Dieser Vorgang kann je nach Rechner einige Zeit in Anspruch nehmen.

Der eigene Modellordner

AeroFly5 bietet die Möglichkeit, Modelle nicht nur im aircraft Ordner unter AeroFly5, sondern auch in einem eigenen Ordner abzulegen. Basis für alle Beispiele in dieser Anleitung ist ja das Modell PT40, und daher legen wir uns einen eigenen Ordner mit dem Namen pt40test für dieses Modell an und gehen rezeptartig wie folgt vor:

- Anlegen eines eigenen Modellordners unter <...\myaerofly\aircraft\pt40test>. Die Punkte stehen für das Laufwerk und ggf. weitere Verzeichnisse oberhalb von myAeroFly5. Aus Kompatibilitätsgründen mit der MAC Version verwenden wir für die Ordnerbezeichnung im aircraft Ordner nur Kleinbuchstaben.
- Öffnen der Datei main.mcf aus dem AeroFly5 Unterverzeichnis config, mit einem Texteditor. Suchen des Textes [user_folder] und Ändern des Parameters in [myaerofly5]. Der Parameter ist der Wert in den eckigen Klammern nach dem Suchtext. Wenn noch nie ein eigener Modellordner angelegt wurde, steht nichts zwischen den Klammern. Dieser Ordner wird in dieser Anleitung ab jetzt „myModellOrdner“ genannt. Liegt der Ordner auf dem Laufwerk C: im Verzeichnis Work sieht die Zeile so aus:

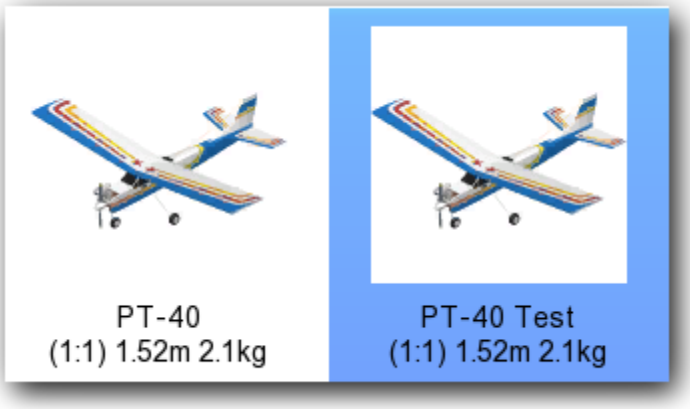
  <[string8][user_folder][c:\work\myaerofly5\]>

- Im Normalfall, also wenn wir ein Modell komplett neu erzeugen wollen, kopieren wir jetzt als Basis für das weitere Vorgehen die TMC und TMD Dateien eines vorhandenen, möglichst ähnlichen Modells in den myModellOrdner. In unserem Fall können wir einfach alle Dateien des Ordners „pt40“ einfügen. Wichtig beim Anlegen von neuen Modellen ist, dass alle Parameterdateien im aircraft Ordner den gleichen Namensanfang haben wie der Ordnername. Da unser Ordner pt40test heißt, müssen jetzt also noch die TMD, die TMC und die TGC Datei in <pt40test.tmd>, < pt40test.tmc> und <pt40test.tgc> umbenannt werden. Damit das Modell auch in der Modellliste als unser Testmodell zu erkennen ist, geben wir ihm auch noch einen eigenen Namen. Dazu öffnen wir die Datei <pt40test.tmc> und editieren den Wert von [DisplayName] zu [PT 40 Test].

Für die Erzeugung eines komplett eigenen Modells gilt zusätzlich:

- Das Modell ist fertig gezeichnet, alle Anforderungen an die Zeichnung sind erfüllt.
- Kopieren der aus der Zeichnung konvertierten Datei mit der Endung TGC in den myModellOrdner.
- Kopieren der konvertierten BMP Dateien mit der Endung TTX in den myModellOrdner.
- Kopieren der TMC und TMD Dateien eines vorhandenen, möglichst ähnlichen Modells in den myModellOrdner.

Jetzt sind alle notwendigen Dateien in unserem Ordner und wir können den Simulator starten und uns das Ergebnis schon mal anschauen. In der Modellauswahl stehen jetzt zwei PT40 Modelle zur Verfügung.



Die Parameterdateien TMD und TMC

Die Beschreibung der Eigenschaften eines Modells wird in zwei Dateien untergebracht. Die kleinere TMC Datei beinhaltet im Wesentlichen die Angaben darüber, unter welchem Namen das Modell angezeigt wird, welche Details es über das Modell zu berichten gibt, welchen Maßstab das Modell ggf. hat und um welche Art von Modell (Heli, Fläche,...) es sich handelt. Die TMD Datei hingegen beinhaltet alle Angaben zur Physik, Aerodynamik, Optik und zum Sound des Modells. Diese Datei ist daher naturgemäß wesentlich umfangreicher als die TMC Datei.

Beide Dateien befinden sich immer im Verzeichnis des jeweiligen Modells, zum Beispiel <AeroFly5/aircraft/modellname/modellname.tmd>, oder entsprechend im myModellOrdner.

Diese beiden Dateien enthalten alle wesentlichen Angaben über die Eigenschaften des Modells bzw. Angaben über die Verbindung zwischen der 3D Zeichnung und der Darstellung im Simulator. Als Textdateien sind sie menschenlesbar und einfach zu bearbeiten, erzwingen jedoch auch ein gewisses Maß an Konventionen in der Schreibweise. Beim Start des Simulators werden diese Dateien eingelesen und deren Inhalt interpretiert. Das kann nur funktionieren, wenn alle Angaben innerhalb der Dateien eindeutig sind. Die TMC und TMD Dateien unterliegen daher in bestimmten Bereichen einer festen Nomenklatur, das heißt, bestimmte Formatierungen und Schreibweisen müssen zwingend eingehalten werden.

Erklärende Texte innerhalb der Parameterdateien sind in dieser Anleitung wie Zeilenkommentare in der Programmiersprache C mit // dargestellt.

Datentypen und Wertebereiche

In den Parameterdateien werden eigene Datentypen verwendet. Diese sind an die in C übliche Schreibweise angelehnt und meist selbsterklärend. Hier die am häufigsten verwendeten Typen und deren Wertebereich.

```
[string8]             // Text
[string8array]        // Text Array
[bool]                // true / false
[int32]               // Ganzzahl mit 4 Byte
[float32]             // Gleitzahl mit 4 Byte
[float32array]        // Gleitzahl Array mit 4 Byte pro Wert
[float64]             // Gleitzahl mit 8 Byte
[float64array]        // Gleitzahl Array mit 8 Byte pro Wert
[tmvector3d]          // Vektor (XYZ)
[tmmatrix3d]          // Matrix (XYZ XYZ XYZ)
```

Aufbau einer TMC Datei

Die TMC setzt sich aus mehreren Parametern zusammen, die wiederum zu einer Datei zusammengefasst werden.

```
<[file][][]                 // Kennung, dass es sich um eine Datei handelt
...                         // Parameter der TMC Datei
>
```

- In der gesamten TMC Datei dürfen keine TABs vorkommen. Einrückungen erfolgen immer mit 4 Leertasten (Spaces).
- Blöcke, Objekte und deren Eigenschaften werden in spitzen Klammern „< >“ zusammengefasst und geschachtelt. Namen, Bereiche und Werte werden in eckigen Klammern „[ ]“ angegeben.
- Alle Werte haben Standardeinheiten wie Kg, Meter, Rad, N oder Hz. Die Einheit selbst wird nicht explizit angegeben.
- Als Nachkomma-Trennzeichen wird der Punkt verwendet.
- Eventuell verwendete Zeilenkommentare (wie in C mit // gekennzeichnet) sind vor der Veröffentlichung des Modells aus der TMC Datei zu entfernen.

Aufbau einer TMD Datei

Nomenklatur-Pflichtteil

Wie erwähnt muss die TMD Datei vom Simulator gelesen und verstanden werden können. Abweichungen von der vorgegebenen Schreibweise werden daher vom Simulator mit einem Absturz des Programmes gleich beim Start quittiert. Wenn der Simulator also mal nicht startet, lohnt es sich als Erstes alle der folgenden Vorgaben zu prüfen.

Die TMD setzt sich aus drei Blöcken zusammen, die wiederum zu einer Datei zusammengefasst werden.

```
<[file][][]                 // Kennung, dass es sich um eine Datei handelt
    <[int32][dynamics][0]   // Kennung für Bereich dynamische Eigenschaften
    ...
    >
    <[int32][graphics][0]   // Kennung für Bereich graphische Eigenschaften
    ...
    >
    <[int32][sound][0]      // Kennung für Bereich Sound Eigenschaften
    ...
    >
>
```

- In der gesamten TMD Datei dürfen keine TABs vorkommen. Einrückungen erfolgen immer mit 4 Leertasten (Spaces).

- Blöcke, Objekte und deren Eigenschaften werden in spitzen Klammern „< >“ zusammengefasst und geschachtelt. Namen, Bereiche und Werte werden in eckigen Klammern „[ ]“ angegeben.
- Alle Werte haben Standardeinheiten wie Kg, Meter, Rad, N oder Hz. Die Einheit selbst wird nicht explizit angegeben.
- Als Nachkomma-Trennzeichen wird der Punkt verwendet.
- Eventuell verwendete Zeilenkommentare (wie in C mit // gekennzeichnet) sind vor der Veröffentlichung des Modells aus der TMD Datei zu entfernen.
- Namen setzen sich grundsätzlich aus einzelnen englischen Worten zusammen. Die einzelnen Wortteile beginnen mit Großbuchstaben; Bsp.: LowerLeftGear. (Hintergrund: Im Modelleditor werden diese Namen radebrechend ins Deutsche übersetzt. Die Verwendung von umgangssprachlichen Elementen würde dies unterbinden.)

Nomenklatur-Kür

Neben den erwähnten, zwingend notwendigen, Schreibweisen, gibt es noch den Bereich, der es dem Menschen leichter macht, eine TMD Datei zu lesen. Die folgenden Angaben sind daher als Empfehlungen zu sehen.

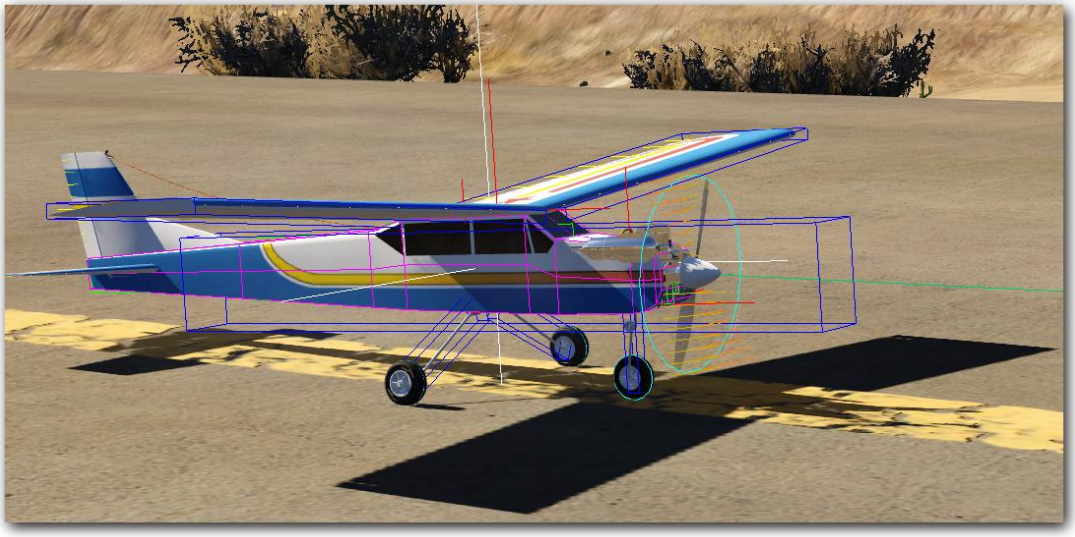
- Alle Float-Werte sollen mit 6 Nachkommastellen angegeben werden. Gegebenenfalls muss mit Nullen aufgefüllt werden.
- Die Einhaltung einer bestimmten Top-Down Reihenfolge bei den Objekten hat sich als vorteilhaft erwiesen. Eine von AeroFly5 gespeicherte Datei dient dabei als Vorlage. Wenn man sich an diese Reihenfolge hält, dann ist es leicht, seine eigene Datei mit der von AeroFly5 erstellten Datei zu vergleichen. Das ist besonders dann wichtig, wenn man den Modelleditor von AeroFly5 dazu benutzt, zum Beispiel Motorkennlinien zu erzeugen.
- Alle Objekte sollten mit sinnvollen Namen versehen werden. Sinnvoll ist in aller Regel die Funktion des Bauteils, eventuell kombiniert mit dem Anbringungsort am Modell; Bsp.: LeftWing, RightStabilizer, LowerFrontGear, LinkRudder,...

Teil 1, Die „Denkweise“ des Simulators

Um ein Modell am Bildschirm darstellen zu können und diesem möglichst realistische Reaktionen auf Steuereingaben des Piloten und der simulierten Umgebung zu geben, bedarf es einiges an Berechnungen. Im Simulator werden in sehr kurzen Zeitabständen sehr viele neue Werte für das jeweilige Modell berechnet. Die Ergebnisse stellen dann jeweils den Zustand des Modells für einen Bruchteil einer Sekunde dar. Einen Teil dieser Berechnungen führt die CPU des Rechners aus, ein anderer Teil wird von der CPU der Graphikkarte übernommen. Je höher die Rechenleistung, besonders die der Graphikkarte, umso flüssiger bzw. umfangreicher kann die Darstellung werden. Für die folgenden Betrachtungen ist es unerheblich, in welchem Teil des Rechners welche Berechnung durchgeführt wird. Hier geht es ausschließlich darum, wie der Simulator die Parameterwerte interpretiert und für seine Berechnungen heranzieht.

Wie ein echtes Modell besteht das Simulator Modell aus verschiedenen Teilen. Diese wurden in der 3D Zeichnung bereits angelegt und mit Namen versehen. Zur Veranschaulichung verwenden wir die PT40, ein einfaches Flächenmodell mit Verbrennungsmotor. In der Zeichnung sind die Teile Rumpf (Fuselage), linker und rechter Flügel (LeftWing, RightWing) mit den Querrudern (LeftAileron, RightAileron), Höhenleitwerk (Stabilizer) mit Höhenruder (Elevator), Seitenleitwerk (Rudder), Motor (Engine), Propeller, Fahrwerk (LeftGear, RightGear, FrontGear), Räder (LeftWheel, RightWheel, FrontWheel), das Höhenruder Servo (ElevatorServo) sowie das Höhenruder Servohorn (ServoArmElevator) und die Verbindung dazwischen (ElevatorLinkage) gezeichnet. Jedes dieser Teile stellt für den Simulator rechnerisch eine eigene Einheit dar. Es gilt nun, die Teile dem Simulator in dynamischer-, optischer- und soundmäßiger Weise bekannt zu machen. Dies geschieht mit Hilfe von sogenannten Objekten, die in Parameterdateien beschreiben werden und die je nach Objekt untereinander verknüpft sein können oder müssen.

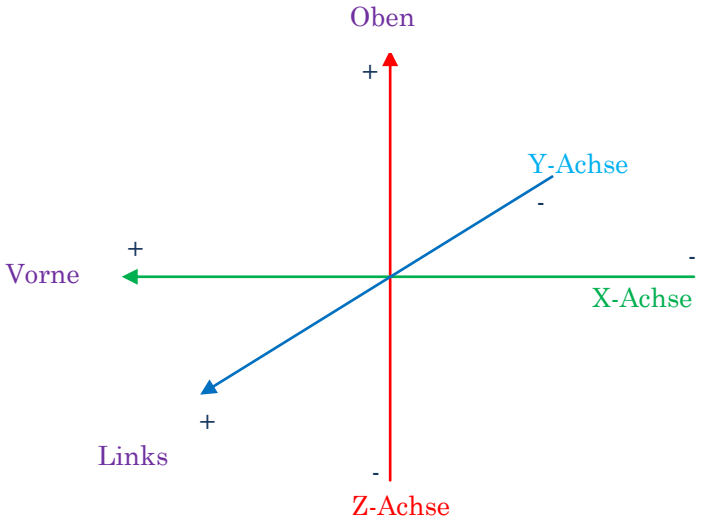
Im AeroFly5 können einige dieser Objekte über die Tastenkombination <Strg F12> als „Physics Info“ angezeigt werden.



Die flugfertige PT40 mit aktivierter Physics Info. Welche Bedeutung die einzelnen Linien und Farben haben werden wir im weiteren Verlauf erfahren.

Das Koordinatensystem

Da das Modell natürlich eine Ausdehnung bzw. eine Größe hat, muss ein 3 dimensionales Koordinatensystem festgelegt werden, mit dem alle Körper, Achsen und Punkte räumlich zu bestimmen sind. Dieses Koordinatensystem wurde bereits bei der 3D Zeichnung angelegt. Sein Nullpunkt liegt üblicherweise nahe des gedachten Modellschwerpunktes, also zentral im Rumpf. In Aerofly5 stellt die X-Achse die Längsachse des Modells dar. Positive Werte sind vor dem Nullpunkt, negative dahinter. Die Y-Achse entspricht der Querachse. Hier gilt: Positive Werte zeigen nach links, negative Werte nach rechts. Schließlich stellt die Z-Achse die Hochachse dar und hier sind positive Werte oberhalb des Nullpunktes und negative Werte unterhalb.



Über diese Koordinaten lassen sich recht einfach Punkte und rechtwinklig angeordnete Achsen und Körper im Raum festlegen.

Das Modell im Simulator

Wie bereits erwähnt, „sieht“ der Simulator das Flugmodell als physikalisches (dynamisches) und als optisches bzw. graphisches Modell. Hinzu kommt dann noch der Sound, der vom Modell oder dessen Komponenten erzeugt wird; hier kann aber nicht von einem eignen Modell gesprochen werden. Interessant und wichtig zum Verständnis ist die Trennung in Dynamik und Optik. Die Physik bestimmt alleine und ausschließlich das Verhalten des Modells und dabei ist es zunächst einmal völlig egal, wie dieses aussieht. Die Optik andererseits kümmert sich ausschließlich um die Darstellung des Modells, ihr sind dynamische Vorgänge fremd. Verbindungen zwischen den beiden gedachten Modellen gib es natürlich schon. So muss beispielsweise einem physikalischen Objekt, das für den Kontakt mit der Umgebung verantwortlich ist, auch eine Hülle gegeben werden, die den tatsächlichen Ausdehnungen des Modells entspricht und die bei Kontakt mit der Umgebung reagiert.

Die Physik - das dynamische Modell

Kräfte im Simulator

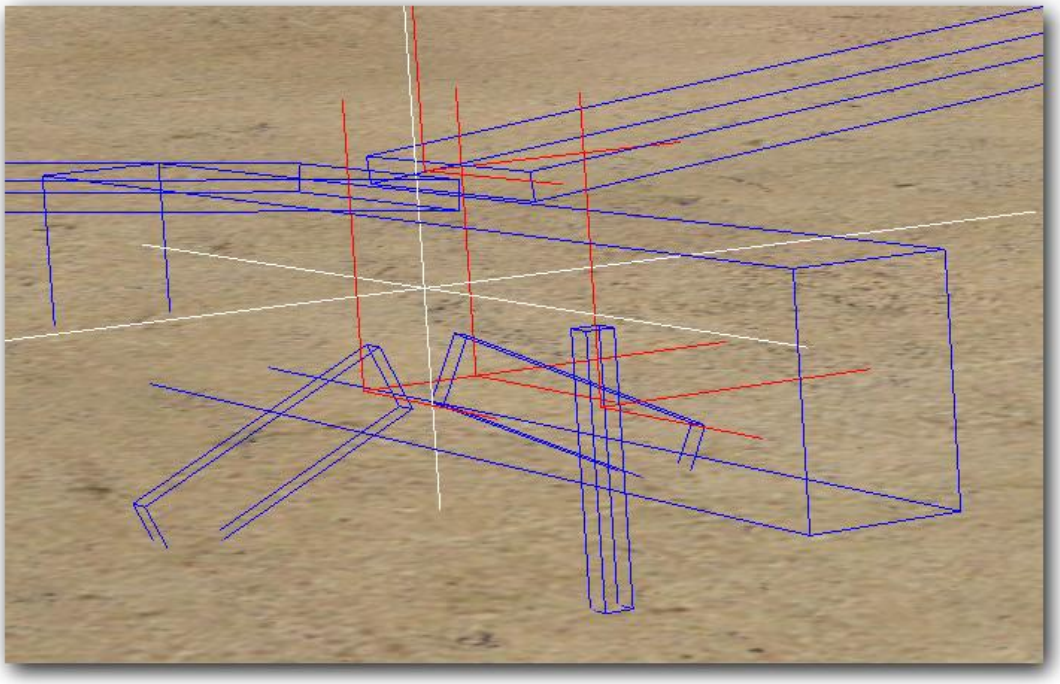
Wenn wir ein reales Flugmodell im Flug betrachten, so wirken hier vielfältige Kräfte auf die verschiedenen Teile des Modells ein. Die umgebende Luft muss das Profil umströmen und erzeugt dabei einen Widerstand und natürlich den Auftrieb. Der Auftrieb wiederum bewirkt eine Kraft, die den Flügel nach oben bewegt und diese Kraft wirkt auch auf die Verbindung zwischen Rumpf und Fläche. Wird hohe Fahrt schnell in Höhe umgesetzt, so wirkt ein Vielfaches der Rumpfmasse ebenfalls als Kraft auf die Verbindung zwischen Rumpf und Fläche. Ist diese Kraft höher als die Haltekraft der Verbindung zwischen Rumpf und Fläche, bricht diese ab. Ebenso zieht das Modell den Kürzeren, wenn es unsanft mit dem Boden in Berührung kommt. Hier wirken in sehr kurzer Zeit hohe Kräfte, weil die gesamte Masse des Modells innerhalb von Sekundenbruchteilen abgebremst wird.

Es gibt also die durch die umgebende Luft induzierten Kräfte, wie Luftwiderstand und Auftrieb und es gibt Kräfte, die, physikalisch gesehen, einem Feder-Dämpfer-System entsprechen. Für die erste Kategorie sind die aerodynamischen Eigenschaften des Modells (Flächenprofil, Rumpfquerschnitt,…) maßgebend, für die zweite Kategorie sind es die Massen und Ausdehnungen. Insbesondere solche Feder-Dämpfer-Systeme werden uns bei der Parametrisierung des Modells häufiger begegnen. Sehr anschaulich wird dieses Thema im Tutorial Physics Lab von Dr. Marc Borchers dargestellt und erklärt. Es lohnt daher, einen genaueren Blick darauf zu werfen. Jedes Feder-Dämpfer System in AeroFly5 wird durch die Federkraft und Dämpfungswerte in alle Richtungen definiert. Die passenden Parameter für die Federkräfte sind Kfx, Kfy und Kfz und für die Dämpfung Dfx, Dfy und Dfz. Die Festigkeit einer Verbindung lässt sich so individuell in allen Richtungen beliebig definieren.

Körper und Verbindungen

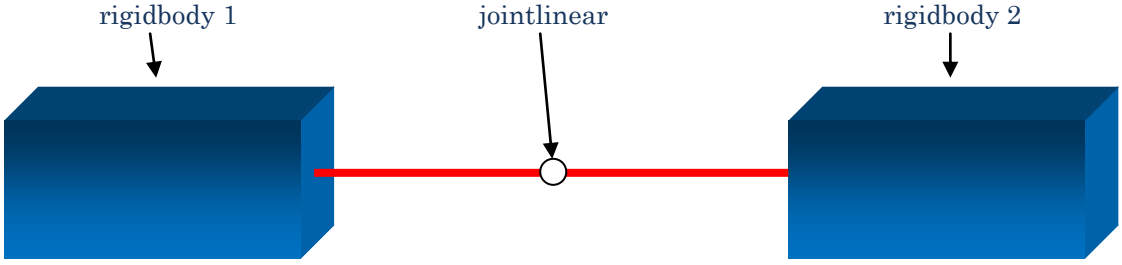
Für die Berechnung der Dynamik des Modells werden zunächst einfache Quader (Objektname: rigidbody) um die graphische Hülle (Ausdehnung) gelegt. Mit Hilfe von Verbindungen (Joints) werden diese Körper dann quasi zusammengeklebt. Der Rumpf bildet dabei die Zentrale. An ihm sind die Flügel befestigt und an diesen ggf. weitere Anbauteile wie zum Beispiel das Fahrwerk. Jeder Quader hat im Wesentlichen eine Ausdehnung, eine Masse und eine Lage im Raum. Damit kann dessen Reaktion auf

Bewegungsänderungen und die damit erzeugten Kräfte, die dann auf den Körper und die damit verbundenen anderen Körper wirken, ermittelt werden. Ein rigidbody hat zwar Masse und Ausdehnung und daher Massenträgheit, er hat jedoch noch keinerlei Bezug zur Umwelt. Er würde daher im Simulator einfach durch den Boden fallen. Erst wenn wir ihm einen Bezug zur Graphik geben, „weiß“ der Simulator welche Ausdehnung vorliegt und kann den freien Fall am Boden stoppen. Als Objekt stehen uns dafür sogenannte Kollisionshüllen (collisionhull) zur Verfügung, auf die ich gleich noch eingehe. Ist ein rigidbody mit einer collisionhull versehen und prallt auf einen festen Widerstand (Boden, Hindernis), so entstehen schnell Kräfte, die höher sind als die für die Verbindungen zwischen den Körpern definierten Werte. Prallt ein solcher Körper auf einen festen Widerstand (Boden), so entstehen schnell Kräfte, die höher sind, als die für die Verbindungen zwischen den Körpern definierten Werte. Die Verbindung bricht, oder wird, wenn sie elastisch genug ist, verbogen. Damit haben wir bereits die entscheidenden Merkmale von Körpern (rigidbody) und Verbindungen (joint) kennengelernt.



Die PT40 hier ohne jegliche Graphik. Rumpf, Flächen und Fahrwerk sind als ridigbody parametrisiert und über jointlinear Objekte verbunden. Die blauen Quader stellen die rigidbodies dar, die Ursprünge der roten Linien zeigen die Verbindungspunkte.

Zur weiteren Veranschaulichung hier nochmal ein einfaches Modell aus zwei rigidbodies und einem jointlinear



Die Verbindung können wir uns als den „Klebepunkt“ vorstellen, der zwei gedachte Gestänge, die eine Linie zwischen den Mittelpunkten der rigidbodies bilden, zusammenhält. Auch hier ist ein Feder-Dämpfer System am Werk. Zusätzlich können die Körper gegeneinander in mehreren Achsen über Servos gedreht werden. Diese Eigenschaft wird zum Beispiel bei lenkbaren oder einziehbaren Fahrwerken benötigt.

Schauen wir uns das gleich mal in der Praxis an. In der Datei <pt40test.tmd> suchen wir den jointlinear, der den linken Flügel mit dem Rumpf verbindet. Wenn die Namensgebung richtig angewandt wurde, müsste dieser [JointFuselageLeftWing] heißen.

```
<[string8][object][jointlinear]                    // Objekt Typ
    <[string8][Name][JointFuselageLeftWing]>       // Name des Objekts
    <[string8][Body0][Fuselage]>                   // rigidbody Bezug 0
    <[string8][Body1][LeftWing]>                   // rigidbody Bezug 1
    <[tmvector3d][R0][-0.020000 0.000000 0.100000]> // Position im Raum
    <[tmvector3d][X0][1.000000 0.000000 0.000000]>  // Lage im Raum
    <[tmvector3d][Y0][0.000000 1.000000 0.000000]>  // Lage im Raum
    <[tmvector3d][Z0][0.000000 0.000000 1.000000]>  // Lage im Raum
    <[float64][Kfx][10000.000000]>                 // Federkraft X Achse
    <[float64][Kfy][10000.000000]>                 // Federkraft Y Achse
    <[float64][Kfz][10000.000000]>                 // Federkraft Z Achse
    <[float64][Dfx][2.000000]>                     // Dämpfung X Achse
    <[float64][Dfy][2.000000]>                     // Dämpfung Y Achse
    <[float64][Dfz][2.000000]>                     // Dämpfung Z Achse
    <[float64][Ktx][1000.000000]>          // Federkraft (Torsion) X Achse
    <[float64][Kty][1000.000000]>          // Federkraft (Torsion) Y Achse
    <[float64][Ktz][1000.000000]>          // Federkraft (Torsion) Z Achse
    <[float64][Dtx][2.000000]>             // Dämpfung (Torsion) X Achse
    <[float64][Dty][2.000000]>             // Dämpfung (Torsion) Y Achse
    <[float64][Dtz][2.000000]>             // Dämpfung (Torsion) Z Achse
    <[float64][ForceMax][100.000000]>      // Maximale Haltekraft
    <[float64][TorqueMax][100.000000]>     // Maximale Kraft bei Torsion
    <[tmvector3d][Rotation0Axis][1.000000 0.000000 0.000000]> // Lage Achse0
    <[float64][Rotation0Angle][0.000000]>                     // Drehwinkel Achse0
    <[tmvector3d][Rotation1Axis][1.000000 0.000000 0.000000]> // Lage Achse1
    <[float64][Rotation1Angle][0.000000]>                     // Drehwinkel Achse1
    <[tmvector3d][PreLoad][0.000000 0.000000 0.000000]> // Vorspannung
    <[tmvector3d][PreTension][0.000000 0.000000 0.000000]> //Vorspg. Torsion
>
```

Das Objekt jointlinear mit dem Namen JointFuselageLeftWing der pt40. Die gelb hinterlegten Kommentare sind in der Datei im Auslieferungszustand natürlich nicht enthalten.

Als erstes Ändern wir mal die Federkraft in der X Achse von 10000 auf 100.

```
Original:          <[float64][Kfx][10000.000000]>
Geänderte Zeile:   <[float64][Kfx][10.000000]>
```

Solange das Modell am Boden steht, scheint noch alles normal zu sein. Bei genauer Betrachtung sieht man jedoch bereits, dass der linke Flügel leicht nach hinten verschoben ist. Spätestens im Flug wird dann deutlich, was wir getan haben. Der Flügel gleitet scheinbar auf einer Schiene in X Richtung hin und her. Das ist auch nicht weiter verwunderlich, denn wir haben ja die Federkraft der X Achse massiv geschwächt. Deutlich langsamer geht dieses Gleiten vonstatten, wenn wir die zugehörige Dämpfung

erhöhen, in dem wir [float64][Dfx] von [2.000000] auf [20.000000] setzen. Nachdem alle Werte wieder im Originalzustand sind, probieren wir gleich noch den Parameter Haltekraft aus und verkleinern diesen um den Faktor 10.

```
Original:         <[float64][ForceMax][100.000000]>
Geänderte Zeile:  <[float64][ForceMax][10.000000]>
```

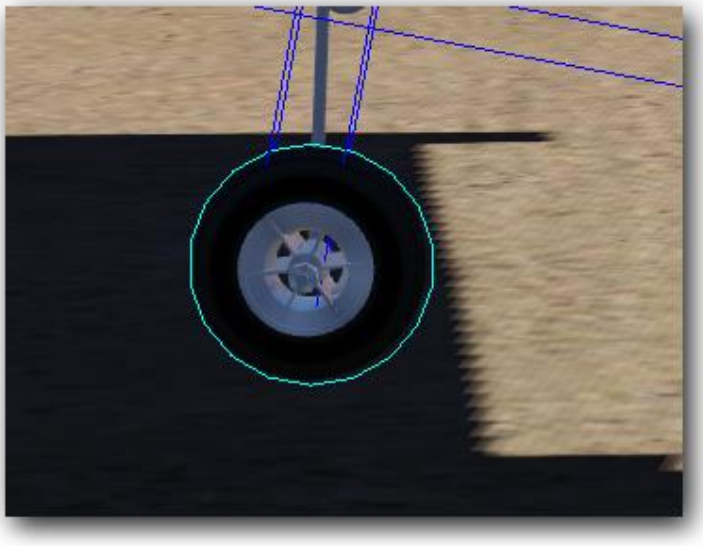
Am Boden sieht wieder alles ok aus, aber bereits beim Versuch abzuheben, reißt der linke Flügel ab. Kein Wunder, haben wir doch gewaltig am „Leim“ gespart. Sollte jetzt jemand auf die Idee kommen, die Werte einfach beliebig hoch einzustellen, um ein für alle Mal sicher zu stellen, dass eine solche Verbindung niemals bricht oder nachgibt, der ist auf dem falschen Weg. Die Kunst ist es, diese Werte so einzustellen, dass sie dem realen Modell entsprechen. Je nach Modell bricht nun mal ein Flügel ab, wenn ich mit hoher Fahrt voll Nick ziehe, oder bei Vollgas extreme Rollraten fliege.

Kontakt mit der Umgebung

Beim Berühren von Boden und Hindernissen wirken Kräfte auf ein reales Modell ein. Die geschieht im Simulator mit Hilfe von collisionhulls für die rigidbodies und mit wheelhulls für die Räder. Solche Kollisionshüllen haben immer ein gezeichnetes Objekt (Rumpf, Fläche,...) als Referenz. Sie „umschließen“ quasi den Körper und erzeugen dann Kräfte, wenn eine Berührung erfolgt. Bei Körpern ist normalerweise ein Crash die Folge, Räder haben bei Bodenkontakt eine Reibung und werden folglich um ihre Achse gedreht.

Ein weiteres Objekt aus dieser Klasse ist collisionblade, die Kollisionshülle für den Propellerkreis. Sie sorgt dafür, dass eine Aktion gestartet wird, sobald der Propeller etwas berührt. In der Regel wird diese Aktion den Motor stoppen.

Schließlich muss unser Modell auch noch irgendwie eine Startposition bekommen, denn wenn der Simulator gestartet wird, oder nach einem Absturz, muss das Modell ja definiert platziert werden. Dazu dienen die Kontaktangaben ContactPoint, ContactSize und ContactNumber in der TMC Datei. Hiermit kann genau festgelegt werden, wo das Modell aufgesetzt wird.



Die wheelhull wird mit aktivierter Physics Info angezeigt. Sie wird als cyan- farbiger Ring dargestellt und sollte das Rad gut umschließen.

Zur Veranschaulichung kommentieren wir einfach die wheelhull des Vorderrades aus. Wir suchen also nach dem Objekt [wheelhull] mit dem Namen [FrontWheelHull] und stellen jeder Zeile des Objekts ein // voran.

```
// <[string8][object][wheelhull]
    // <[string8][Name][FrontWheelHull]>
    // <[float64][X][5000.000000]>
    // <[float64][D][2.000000]>
    // <[float64][Radius][0.033000]>
    // <[string8][Body][FrontGear]>
    // <[tmvector3d][R0][0.200000 0.000000 -0.148000]>
    // <[tmvector3d][X0][1.000000 0.000000 0.000000]>
    // <[float64][RollingFrictionCoefficient][0.070000]>
    // <[float64][BrakeCoefficient][1.000000]>
// >
```

Jetzt hat dieses Rad keine Beziehung zur Umwelt mehr und es versinkt im Boden. Der Propellerkreis berührt den Boden und sein collisionblade Objekt sorgt dafür, dass der Propeller zerstört wird. Das Modell sinkt weiter in den Boden und schließlich sorgt die collisionhull des Rumpfes dafür, dass nicht das ganze Modell vom Erdboden verschluckt wird. Sollte also mal ein Modell so gar nicht auf dem Boden stehen wollen, sondern immer darin versinken, so ist sehr wahrscheinlich eine fehlende collisionhull dafür verantwortlich

Einbauteile

Bislang haben wir uns ausschließlich mit Körpern und deren Dynamik beschäftigt. Natürlich braucht unser Modell auch eine Fernsteuerung mit allen Komponenten, einen Motor und evtl. etwas Zubehör.

Zur Fernsteuerung gehören der Empfänger (receiver), die Empfängerausgänge (receiveroutput) und die Servos (servolinear oder servoclassic). Beim Empfänger bestimmen die angegebenen Funkionen die Anzahl der Steuerkanäle. Jeder Empfängerausgang wiederum stellt den Eingang einer Funktion dar. Diese Verknüpfung finden wir bei den Servos und bei Zusatzfunktionen wieder. Hier sind die Ausgangswerte des Empfängers die Steuerwerte. Ein solcher Ausgang kann mehrere Servos oder Funktionen steuern. Die Servos selbst haben je nach Typ vielfältige Möglichkeiten der Beeinflussung. Die Laufrichtung, Laufgeschwindigkeit und die Laufkurve können ebenso bestimmt werden wie die Neutrallage und der maximale Ausschlag.

Nehmen wir uns zum praktischen Test das linke Querruder Servo vor. Wir finden es als [servoclassic] mit dem Namen [ServoLeftAileron]. Wenn wir hier die Steigung wie in der Abbildung von 0.400000 auf -0.400000 ändern, läuft dieses Servo genau invertiert.

```
<[string8][object][servoclassic]          // Objekt Typ
    <[string8][Name][ServoLeftAileron]> // Objekt Name
    <[string8][Input][AileronInput.GetOutput]> // Steuereingang
    <[float64][Speed][2.500000]>     // Laufgeschwindigkeit
    <[float64][P0][0.000000]>        // Nullage Knüppel Mitte
    <[float64][P1][-0.400000]>       // Steigung (Maxwert)
    <[float64][P2][0.000000]>        // Expo1
    <[float64][P3][0.000000]>        // Expo2
    <[float64][Position][0.000000]> // Lage nach Restart
>
```

Das linke Querruder geht bei Knüppel links statt nach oben nach unten. Da jetzt bei Querruder links beide Ruderflächen nach unten gehen, ist zu erwarten, dass sich das im Flug wie ausgefahrene Klappen auswirkt. Entsprechend müsste Querruder rechts wie Störklappen wirken. Dass das genau so simuliert wird, zeigt ein Testflug, bei dem die PT40 über Höhe und Seite gesteuert wird. Mit dem Querruder können wir den Auftrieb vergrößern (Knüppel links) und verkleinern (Knüppel rechts).

Wenn wir keinen reinen Segelflieger modellieren möchten, brauchen wir einen Motor als Antrieb. In der Simulation gibt es dafür die Objekte engineelectric (Elektromotor) und enginebasic (Verbrennungsmotor). Ein Motor erzeugt ein Drehmoment, welches über eine Welle (driveshaft) auf den Propeller wirkt. Auf diesen wirkt der Luftwiderstand und es stellt sich eine Drehzahl an der Welle ein, die aus dem Drehmoment des Motors und dem Widerstand am Propeller resultiert. Die Welle selbst hat dabei keine Parameter, sie ist lediglich das Koppelglied zwischen Motor und Propeller.

Als Vertreter der Zusatzfunktionen sei hier der Rauchgenerator genannt. In der Simulation gibt es dafür das dynamische Objekt smoke und das graphische Objekt smokegraphics. Mit diesen Objekten kann sowohl der normale Motorqualm des Verbrennungsmotors dargestellt werden, als auch ein über eine Steuerfunktion zuschaltbarer Effektrauch.

Effektrauch an der Adrenalin 120

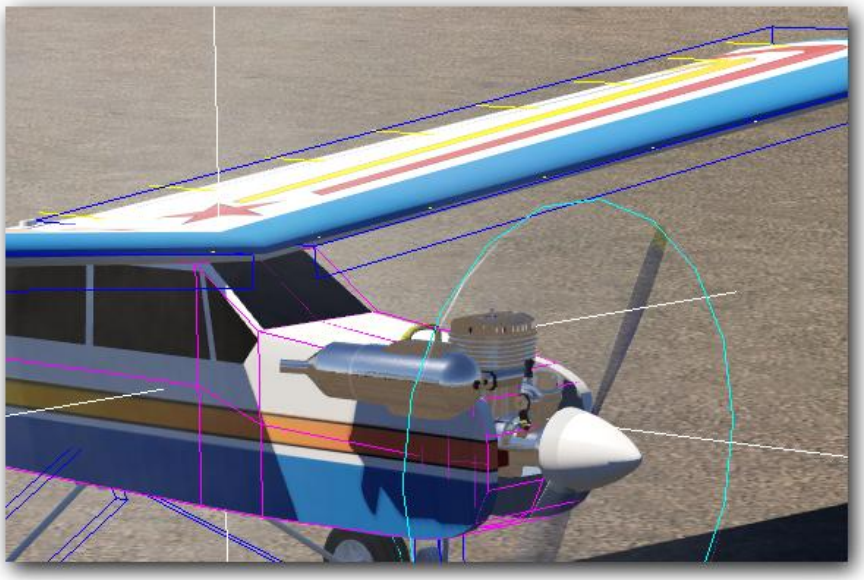
Aerodynamik

Wesentlich verantwortlich dafür, dass ein Modell in der Luft fliegen kann, ist das Profil der Flügel. Es erzeugt, wenn es angeströmt wird, die Auftriebskraft, aber natürlich auch den Luftwiderstand. Mit dem Objekt airfoil wird für die einzelnen Tragflächen das Profil definiert. Im Falle des Hauptflügels muss das Profil sogar für Flügelwurzel und Flügelspitze separat definiert werden. Das Profil sagt jedoch noch nichts über Größe, Form und Aufbau des Flügels aus. Hierfür ist das Objekt aerowing zuständig. Hiermit wird in Schnitten angegeben, welche Ausdehnung und welche Lage im Raum der Flügel hat und ob eventuell Klappen oder Ruder angebracht sind. Wichtig dabei ist, dass diese Schnitte immer von rechts nach links angegeben werden. Am linken Flügel also vom Rumpf beginnend und am rechten Flügel an der Flügelspitze beginnend. Immer da, wo sich an der Ausdehnung oder am Aufbau (Klappen, Ruder) des Flügels etwas ändert, muss ein Schnitt angelegt werden. Der folgende Ausschnitt der Parameter des Objektes [aerowing] mit dem Namen [LeftWingAero] zeigt beispielhaft diese Schnitte.

```
[StationY][0.000000 0.091000 0.740000 0.774000 ]>     // Position des Schnittes auf der Y Achse
[StationLE][0.072000 0.072000 0.072000 0.072000 ]>    // X Wert Flügennase (Leading Edge)
[StationTE][-0.181000 -0.181000 -0.181000 -0.181000 ]>// X Wert Flügelhnterkante (Trailing Edge)
[StationZ][0.100000 0.108000 0.166000 0.170000 ]>     // Position des Schnittes auf der Z Achse
[StationIncidence][0.000000 0.000000 0.000000 0.000000 ]> // Anstellwinkel an der Schnittposition
[StationFlap][0 1 0 0 ]>                              // Anzeige Ruder / Klappe im Schnittbereich
```

Auch der Rumpf hat aerodynamischen Einfluss. Seine Frontfläche bremst das Modell und im Messerflug kann der Rumpf sogar den kompletten Auftrieb übernehmen, um das Modell in der Luft zu halten. Das Objekt zur Beschreibung der Rumpfaerodynamik heißt aerofuselage. Auch hier wird in Schnitten angegeben, welche Ausdehnung der Rumpf hat und wie viel Widerstand und Auftrieb er an den verschiedenen Flächen erzeugt. Ein separates Objekt für das „Profil“ des Rumpfes gibt es nicht. Beim Rumpf werden die Schnitte von vorne nach hinten angegeben. Auch hier sind alle Stellen relevant, an denen sich die Form oder Größe des Schnittes ändert. Das aerofuselage Objekt wird nicht nur für den Rumpf, sondern für alle Teile am Modell verwendet, die aerodynamischen Einfluss haben. Als Beispiel seien hier besonders ausladende Rumpf- oder Flügelanbauten wie Zusatztanks oder Triebwerksverkleidungen genannt.

Ob die Schnitte am Rumpf und an den Flächen gut gewählt wurden, lässt sich leicht anhand der magenta-farbigen Linien am Rumpf und an den gelben Linien in den Flügeln (hier an der Hinterkante des Flügels zu erkennen) feststellen.



Das optische Modell

Feste Teile

Einzeln gezeichnete Teile aus der 3D Zeichnung, die sich zwar im Raum aber nicht gegeneinander bewegen, werden mit dem Objekt rigidbodygraphics dargestellt. Dabei können mehrere Teile zusammengefasst werden. So können mit dem Rumpf auch beispielsweise der Motor und die Leitwerke in der Geometrieliste angegeben werden. Ein rigidbodygraphics Objekt referenziert immer auf die Position und Lage (Orientierung) eines rigidbodies. Mit anderen Worten, wird der rigidbody bewegt, bewegt sich auch die Graphik. Wird in rigidbodygraphics zum Beispiel auf die Position und Lage des rigidbodies eines drehbaren Fahrwerks referenziert, dann bewegt sich auch die Graphik.

```
<[string8][object][rigidbodygraphics]          // Objekt Typ
    <[string8][Name][FrontGear]>               // Objektname
    <[string8][GeometryList][ FrontGear ]>     // Bauteil aus der Zeichnung
    <[uint32][PositionID][FrontGear.R]>        // rigidbody Referenz Position
    <[uint32][OrientationID][FrontGear.Q]>     // rigidbody Referenz Lage
    <[tmvector3d][Shift][-0.210000 0.000000 0.080000]> // Position X,Y,Z
>
```

Hier als Beispiel das Vorderrad der PT40. Das rigidbody Objekt [FrontGear] wird über den [jointlinear] [JointFuselageFrontGear] mit dem Parameter [Rotation0Control] bewegt. Der Wert [ServoNoseWheel.Position] verbindet das Servo mit dem Objekt.

Bewegliche Teile

Eine andere Art der Bewegung stellen Teile dar, die über Scharniere oder Gelenke mit einem festen Teil verbunden sind. Dazu gehören alle Ruder und Klappen, aber auch Cockpithauben, die geöffnet werden können. Sie werden über das Objekt hingedbodygraphics definiert. Wichtig dabei ist es, den Drehpunkt (Pivot) und die Drehachse (Axis) exakt anzugeben. Falsche oder ungenaue Angaben führen dazu, dass Ruder in Flächen unnatürlich einschneiden, oder schräg wegkippen.

```
<[string8][object][hingedbodygraphics]                  // Objekt Typ
    <[string8][Name][LeftAileron]>                      // Objekt Name
    <[string8][GeometryList][ LeftAileron ]>            // Bauteil aus der Zeichnung
    <[uint32][PositionID][LeftWing.R]>                  // rigidbody Referenz Position
    <[uint32][OrientationID][LeftWing.Q]>               // rigidbody Referenz Lage
    <[uint32][AngleID][ServoLeftAileron.Position]>      // Steuerservo
    <[tmvector3d][Axis][0.000000 0.995610 0.093601]>    // Drehachse
    <[tmvector3d][Pivot][-0.137200 0.415400 0.131300]>  // Drehpunkt
    <[float64][AngleMax][0.900000]>                     // Maximaler Ausschlag
>
```

Das hingedbodygraphics Objekt LeftAileron der PT40

Das Vorzeichen des Parameters [AngleMax] bestimmt die Laufrichtung des bewegten Objektes. Falls die Invertierung des linken Querruders noch aktiv ist, können wir jetzt mal das Vorzeichen ändern und -0.900000 probieren. Das Ruder läuft jetzt optisch wieder richtig, die Invertierung der Wirkrichtung besteht jedoch nach wie vor. Wir wissen jetzt also, wie wir Wirkrichtung und optische Laufrichtung eines Ruders separat beeinflussen können, und es ist sicher sinnvoll, jetzt beide Werte wieder auf die Ursprungswert zurück zu setzen.

Verbindungen

Über die hingedbodygraphics können die angegebenen Teile bewegt werden, auch wenn gar nicht ersichtlich ist, wie diese Bewegung zustande kommt. Das kann im realen Modell auch so sein, wenn zum Beispiel ein Querruder innerhalb des Flügels angesteuert wird. Wird ein Ruder über ein sichtbares Gestänge ggf. mit Ruderhorn und Servoarm angesteuert, dann kann das im Simulator mit dem Objekt linkagegraphics realisiert werden. Hier wird das gezeichnete Teil (Gestänge) entweder mit einem festen und einem beweglichen Teil oder mit zwei beweglichen Teilen verbunden. Auch hier ist die genaue Angabe der Drehpunkte (Joint 0,1) wichtig für die realistische Darstellung. Hat ein Gestänge auf der einen Seite ein Ruderhorn und mündet auf der anderen blind im Rumpf, so liegt der Drehpunkt des blinden Endes irgendwo nahe dem Rumpfdurchbruch. Letztlich ist dann noch festzulegen, welcher der Drehpunkte als fest verbunden anzusehen ist. Mit dem Parameter AttachTo kann das der Joint0 (Wert 0), der Joint1 (Wert 1) oder beide (Wert 2) sein. Das Gestänge kann man sich dabei wie ein Gummiband vorstellen, das bei Bewegung gedehnt oder gestaucht wird.

```
<[string8][object][linkagegraphics]                    // Objekt Typ
    <[string8][Name][LinkThrottle]>                    // Objekt Name
    <[string8][GeometryList][ LinkVergaser ]>          // Element aus der Zeichnung
    <[string8][Graphics0][Fuselage]>     // Zeichnungselement 0 an dem die Verbindung befestigt ist
    <[string8][Graphics1][ServoVergaser]>// Zeichnungselement 1 an dem die Verbindung befestigt ist
    <[tmvector3d][Joint0][0.194200 -0.015800 0.017300]> // Position der Befestigung 0
    <[tmvector3d][Joint1][0.287500 -0.019300 0.018800]> // Position der Befestigung 1
    <[uint32][AttachTo][2]>              // Befestigt an: Element 0 oder 1 oder an beiden 2
>
```

Hier die Parameter des Gestänges zur Vergaserklappe.

Räder

Räder können sowohl als feststehende Teile, als auch als drehende Teile dargestellt werden. Eleganter und realistischer ist natürlich die zweite Variante. Das Objekt hierfür ist rotatingbodygraphics. Die Rotation wird dabei über den Ausgangswert [WhellHull.RotationAngle] der speziellen Kollisionshülle [wheelhull] übertragen.

Smoke

Zur realistischen Darstellung eines Modell, besonders natürlich eines Modells mit Verbrennungsmotor, gehört auch der Abgasrauch. Im Simulator gib es dafür das dynamische Objekt smoke und das graphische Objekt smokegraphics. Beide zusammen sorgen für den Abgasrauch, können aber auch, über einen Kanal geschaltet, einen Effektrauch erzeugen. Über eine ganze Reihe von Parametern kann bestimmt werden, wie viel Rauch bei welchen Bedingungen angezeigt werden soll. Dabei spielen auch Motodrehzahl und Propellerwind eine Rolle.

Der Sound

Wie der Abgasruch für das Auge dazu beiträgt, dass die Simulation realistisch aussieht, so sorgt der Sound dafür, dass dieser Eindruck akustisch komplettiert wird. Mit dem Objekt soundengine können mehrere Sounddateien verschiedenen Drehzahlen zugeordnet werden. Diese Sounddateien beinhalten üblicherweise den Klang und die Lautstärke des Motors bei einer bekannten Drehzahl. Im Objekt soundengine werden diese Klänge dann so manipuliert, dass ein nahtloser Übergang über den gesamten Drehzahlbereich entsteht.

Teil 2, Erfassen der Modelldaten

Vorgehensweise, Tipps und Tricks

Wer Modelle für AeroFly5 parametrisieren will, muss sich zwangsläufig auch etwas mit dem Zeichenprogramm befassen, mit dem das Modell gezeichnet wurde. Es ist jedoch nicht nötig, dass man alle Feinheiten kennt, oder gar selbst auch Konstrukteur des Modells ist. In diesem Abschnitt möchte ich etwas aus dem Nähkästchen plaudern und Informationen aus der Praxis weitergeben. Wie immer im Leben: Vieles kann man beliebig kompliziert oder eben auch viel einfacher angehen. Meine Hinweise sind dabei nicht als das Maß der Dinge anzusehen, sondern eher als Grundlage für weitere Ideen. Vielleicht entwickelt sich daraus durch regen Austausch ja bald ein eigenes „Handbuch der Tipps und Tricks“ für den AeroFly5. Ein Forum als Plattform für den Informationsaustausch gibt es ja bereits unter < www.aerofly.com>.

Tipps und Tricks beim Vermessen des Modells in der 3D Zeichnung

Was an Daten aus der 3D Zeichnung zu holen ist, lässt sich beschränken auf Punkte, Achsen, Schnitte und die Namen der Einzelteile. Es hängt natürlich wesentlich vom verwendeten Zeichenprogramm ab, was letztlich wie gemessen werden kann. Punkte lassen sich sicher mit jedem Programm leicht erfassen, bei Achsen sieht es schon schwieriger aus. Oft sind an der Stelle, wo eine Achse gemessen werden soll, keine Kanten von gezeichneten Elementen und man muss sich behelfen. Konkretes Beispiel ist die Drehachse eines Querruders. Diese ist normalerweise nicht die Vorderkante des Ruders. Meist bildet ein Halbkreis, der in einer entsprechenden Mulde des Flügels liegt, die Vorderkante. Die Drehachse liegt somit im Mittelpunkt der Sehne des Halbkreises. Am einfachsten wäre es, eine Hilfslinie an dieser Stelle einzuzeichnen, und von dieser die Maße zu nehmen. Wenn das Zeichenprogramm keine Hilfslinien unterstützt, kann man sich mit zwei Punkten helfen und die Drehachse daraus errechnen. Legt man diese beiden Punkte an die Enden des Ruders, so hat man mit dem Mittelpunkt der Achse auch gleich deren Drehpunkt.

Bei Schnitten hat sich bewährt, an der Stelle des Schnittes ein Hilfsrechteck (Panel) zu zeichnen. Von diesem kann man dann leicht die benötigten Schnittdaten ablesen. Weiterer Vorteil dieser Methode ist, dass man diese Panels mit der Zeichnung abspeichern und so jederzeit wieder nachvollziehen kann, wo ein Schnitt gesetzt wurde. Ungünstig gesetzte Schnitte können leicht verschoben werden und fehlende oder zu viel gesetzte Schnitte lassen sich leicht entfernen oder hinzufügen.

Bei der Namensgebung der 3D Objekte hat es sich bewährt, eine eindeutige, wiederkehrende und englischsprachige Nomenklatur zu verwenden. Bei den Standardteilen ist das noch einfach, ein Rumpf heißt immer Fuselage und ein Flügel wird entweder LeftWing oder RightWing heißen. Setzt man bei anderen Teilen fort, dass immer erst die Seite oder Position und dann die Funktion genannt wird, so heißt ein Fahrwerkesteil FrontGear oder LeftGear. Geht es noch weiter in die Peripherie, dann folgt erst wieder die Position und dann die Funktion. Der untere Teil des Fahrwerkes heißt dann LeftGearLowerSuspension. Damit ist klar, dass es sich um den Teil unterhalb des linken Fahrwerks handelt und dass dieser als Federung gedacht ist. Derart gewählte Namen erleichten beim Parametrisieren enorm die Arbeit.

Ebenfalls im Zusammenhang mit den Namen der Zeichenobjekte sei hier ein Tipp genannt, der unter Umständen viel Zeit sparen kann. Achten Sie peinlich genau auf die Rechtschreibung bei den Namen. Ein Teil das LefttWing in der Zeichnung heißt, wird niemals als LeftWing angezeigt werden. Dass da versehentlich ein doppeltes T nach dem Left ist, fällt erst bei genauem Hinsehen auf. Daher: Wenn mal ein Teil nicht sichtbar wird, obwohl man vermeintlich alle Parameter richtig angegeben hat, dann erst mal die Schreibweise des in Frage kommenden Namens mit der Zeichnung prüfen. Um solche Schreibfehler komplett zu vermeiden, kopiert man am besten den rigidbodygraphic Teil aus der tm.log Datei, welche beim Exportieren aus dem 3D Programm erzeugt wird. Hier sind alle gezeichneten Objekte aufgeführt und können so einfach in die eigene TMD Datei übernommen werden. Auf diese Weise werden auch alle Objekte von Anfang an dargestellt. Später müssen dann alle Objekte, die beweglich oder allgemein nicht fest mit dem Rumpf verbunden sind, in die entsprechenden Graphikobjekte überführt werden.

Tipps und Tricks beim Parametrisieren

Der beste und mit Abstand hilfreichste Tipp gleich zu Beginn dieses Abschnittes in einem Wort: Abschauen! Und gleich den zweiten ebenso kurzen und hilfreichen Tipp hinterher: Probieren!

Am schnellsten und lehrreichsten ist es in aller Regel, ein vorhandenes, möglichst ähnliches Modell zu kopieren und zunächst mal alles auszukommentieren, was nicht sicher oder zwingend zum neuen Modell gehört. Dazu gehören vor allem die dynamischen und optischen Objekte von Teilen, die das neue Modell gar nicht hat. Auch Kollisionshüllen, mit Ausnahme der wheelhulls, können erst mal auskommentiert werden.

Sollte sich bei Änderungen ein Fehler eingeschlichen haben, so quittiert der Simulator dies meist recht heftig durch direkten Absturz. Hat man mehrere Schritte auf einmal vorgenommen, so ist es oft zeitaufwändig nachzuvollziehen, wann der Fehler eingebaut wurde. Nach jeder Änderung ist es daher wichtig, die Daten im Simulator zu prüfen. Dazu einfach das Modell neu laden, und das geht extrem einfach und schnell mit der Tastenkombination Shift F12. Im Gegensatz zum normalen Laden des Modells, wird bei dieser Tastenkombination das Modell nicht auf den Startpunkt gesetzt, sondern es bleibt genau in der Position in der es vor dem Laden war. Das ist besonders hilfreich, wenn man Details anschauen und ändern möchte und dazu das Modell erst in eine entsprechende Position bringen muss.

Eine typische Fehlerquelle ist ein Schreibfehler bei Objekten die auf einen rigidbody referenzieren. Beispiel: Name des rigidbody: Fuselage; Referenzangabe: Fusellage. Die Referenz auf einen rigidbody, den es nicht gibt, führt zum Absturz.

Weitere häufige Fehlerquellen sind die Verletzung der im Abschnitt Nomenklatur (Pflichtteil) aufgelisteten Regeln. Wenn beim Löschen versehentlich eine eckige Klammer mit gelöscht wurde, so fällt das beim Betrachten kaum auf, aber auch hier quittiert der Simulator mit Absturz.

Beim Parametrisieren gibt es keine vorgeschriebene Reihenfolge. Im Folgenden werden weitere Tipps in eine bewährte Reihenfolge gepackt. Es wird immer davon ausgegangen, dass je eine von einem verwandten Modell abgeleitete TMD und TMC Datei verwendet wird.

Um möglichst von Anfang an ein komplettes Modell sehen zu können, kann man alle Namen aus der 3D Zeichnung in die Geometrieliste eines einzigen rigidbodygraphics Objektes schreiben. Mit steigender Differenzierung der Teile entfernt man die Teile aus dieser Liste und fügt sie den Funktionen zu.

Die rigidbodies werden in Größe, Masse und Lage als erstes ans neue Modell angepasst. Die Joints können erst mal so bleiben, es sei denn das Modell explodiert gleich beim Laden. Wenn dem so ist, dann sind die Verbindungen meist zu fest, oder die Masse des Körpers ist zu gering. Es hat sich bewährt, die Joints deutlich zu lockern und ggf. die Masse des Körpers zu erhöhen, auch wenn das im Moment nicht der tatsächlichen Masse entspricht. Die korrekten Werte werden dann später ermittelt, wenn das Modell nahezu fertig ist.

Aus der Zeichnung werden dann die Schnitte für den Rumpf und die Flächen übernommen. Auch hier hilft es, nur die Daten zu ändern, die sich vom kopierten Modell

unterscheiden. Rumpfschnitte werden immer von vorne nach hinten und Flügelschnitte von rechts nach links gelegt. Leitwerksschnitte werden horizontal wie Flügel von rechts nach links und vertikal von unten nach oben gelegt.

Als nächstes legen wir die Funktionen des Modells fest. Die Reihenfolge der Funktionsnamen entspricht dabei den Kanälen der Fernsteuerung. Die Namen der Funktionen sind nicht frei wählbar, sondern als Keywords zu sehen. Throttle, Aileron, Elevator und Rudder sind die Grundfunktionen, Flaps, Wheel-Brake, Retractable-Gear, Smoke, Cockpit und Airbrake sind Beispiele für Sonderfunktion. Jede Funktion bekommt schließlich einen Empfängerausgang.

Wenn wir schon an dieser Ecke sind, dann können wir gleich die Servos einbauen und diesen die Empfängerfunktionen zuweisen. An dieser Stelle ist es noch nicht wichtig, dass die Servos bereits in Laufrichtung, Ausschlag und Geschwindigkeit korrekt angegeben sind. Funktionen, die kein Servo benötigen, können auch gleich zugewiesen werden, wenn das entsprechende Objekt bereits vorhanden ist. Als Beispiel sei hier smoke genannt. Hier kann direkt der Kanal zugewiesen werden, da keinerlei Kurven, Laufrichtungen oder Laufgeschwindigkeiten Einfluss haben. Es gibt nur Ein oder Aus. Solche Zuweisungen sind dann auch gerne Fehlerquellen, wenn die Kanalreihenfolge geändert wird. Alle Servos bekommen den Empfängerausgang, und da der in der TMD üblicherweise beim Empfänger steht, wird dieser auch mit geändert. Funktionen wie smoke bekommen das dann nicht mit und man wundert sich, dass smoke plötzlich zusammen mit einer ganz anderen Funktion aktiviert wird.

Als nächstes kümmern wir uns um den Antrieb. Motor, Welle und Propeller werden parametrisiert. Steht das Modell einigermaßen stabil und mit genügend Abstand auf dem Boden, dann kann auch gleich die Propellerdisk aktiviert und parametrisiert werden. Ansonsten folgt diese, nachdem die Aufsetzpunkte festgelegt wurden. Die Antriebsdaten des Motors befinden sich in einer Tabelle. Natürlich kann man diese von Hand verändern, es ist jedoch deutlich einfacher, hierfür den Modelleditor heranzuziehen. Hier kann die Leistungskurve mit der Maus verändert werden, man kann sofort testen, ob das Ergebnis den Vorstellungen entspricht und man kann vor allem die Werte als neues Modell abspeichern. Mit einem Dateivergleichsprogramm lässt sich die neue Tabelle einfach einbinden.

Auch wenn sich noch keine Ruder bewegen, mit dem so definierten Modell kann man bereits fliegen. Dennoch kümmern wir uns jetzt erst mal um die Optik. Wir legen für jedes Teil, das später auch optisch separat behandelt werden soll, ein rigidbodygraphics Objekt an. Das sind alle Teile, die beim Crash abbrechen können und die insofern auch ein passendes rigidbody Objekt haben, an dem sie sich orientieren können. Üblicherweise sind das der Rumpf mit allen festen Teilen, die Flügel, evtl. die Leitwerke und die Fahrwerksteile.

An den Flügeln und an die Leitwerke werden dann mit hingedbodygraphics die bewegten Teile angebracht, und auch die Räder bekommen ihre rotatingbodygraphics Objekte.

Jetzt ist schon viel mehr zu sehen und wir können in einem Zwischenschritt die Wirkrichtungen, Laufgeschwindigkeiten und Ausschläge der Servos und in Folge auch die sichtbaren Ausschläge und Laufrichtungen der Ruder einstellen.

Wenn unser Modell beim Hinstellen noch hefig springt, dann sind die Aufsetzpunkte in der TMC Datei falsch gewählt. Diese sollten so angepasst werden, dass das Modell sich kaum bewegt, wenn es mit der Leertaste neu positioniert wird.

An dieser Stelle ist das Grundsetup abgeschlossen. Was jetzt noch folgt, ist physikalisches und optisches Finetuning. Die Reihenfolge der kommenden Schritte ist noch flexibler als beim Grundsetup und hängt natürlich auch von den Features ab, die eingebaut werden sollen.

Je nachdem, wie nah unsere Vorlage am neuen Modell war, merken wir beim Fliegen, dass das Verhalten noch nicht dem entspricht, was wir erwarten. Anzupassen sind üblicherweise Parameter wie Rumpfwiderstand (Cdx), Anstellwinkel (StationIncidence) und Auftrieb (ClAlpha).

Sind in der 3D Zeichnung Gestänge für die Ansteuerung von Rudern oder auch dem Vergaser, sowie Ruderhörner oder /und Servoarme gezeichnet, dann werden diese jetzt mit linkagegraphics Objekten zum Leben erweckt.

Jetzt kümmern wir uns noch um die Zusatzfeatures wie Rauch, bewegte Teile wie Canopy oder Airbrake und Haken.

Bislang haben wir noch den Klang des Referenzmodelles gehört. Sollten Sounddateien existieren, die näher am Klang des neuen Modells liegen, so sind diese im Soundbereich anzugeben. Ebenso sollte hier überprüft werden, ob sich am Drehzahlbereich etwas geändert hat. Ggf. müssen Anpassungen vorgenommen werden. Beim abschließenden Test sollte der Lautsprecher also nicht abgeschaltet sein ;-)

DAS PT40 TUTORIAL „HOWTO“

Was ist besser geeignet als der Simulator selbst, um die verschiedenen Zusammenhänge der Objekte und Parameterwerte sichtbar zu verdeutlichen. Da der Simulator die Möglichkeit bietet, von einem Modell mehrere Konfigurationen zu verwalten, liegt es nahe, einfach ein Modell von Anfang an und schrittweise als je eine neue Konfiguration anzulegen. Genau das geschieht im Tutorial PT40 „HowTo“ mit der bereits bestens bekannten PT40. Dieses Tutorial steht unter www.areofly.com zum Download bereit. Wir kopieren aus der ZIP Datei das Verzeichnis <pt40tutorial> in einen unserer aircraft Ordner. Das kann entweder der Standardordner <aircraft> des Simulators sein, oder unser eigener myModellOrdner. Wenn wir den Simulator starten sehen wir ein drittes PT40 Modell mit der Bezeichnung PT-40 Tutorial. Dieses laden wir und sehen jetzt, dass bei den „verfügbaren Konfigurationen“ der Pfeil nach unten aktiv ist und sobald wir diesen anklicken sehen wir, dass 20 weitere PT40 Modelle mit der Bezeichnung „PT-40 HowTo“ erscheinen. Sie sind durchnummeriert von 01 bis 20 und jeweils mit einem weiteren Stichwort versehen. Dieses Stichwort gibt einen Hinweis darauf, was in der entsprechenden Lektion des Tutorials behandelt wird.
Ausgehend von einem absoluten Minimalsystem mit nur einem rigidbody wird in Folge das komplette PT40 Modell aufgebaut. Im Fenster „Modellinformationen“ finden sich kurze Erklärungen, was im Einzelnen sichtbar ist bzw. was veranschaulicht wird und mit welchen Tasten ggf. eine sinnvolle Bedienung erfolgen kann. Diese Hinweise möchte ich hier mit weiteren Infos anreichern und in einigen Fällen den Bezug zur TMD Datei herstellen, indem auf einzelne Parameter weiterführend eingegangen wird.

Die TMD Dateien aller Lektionen beinhalten immer nur die Daten, die für den aktuellen Stand des Modells auch tatsächlich benötigt werden. Durch den Vergleich der TMD Dateien benachbarter Lektionen wird so direkt sichtbar, was neu ist bzw. was sich geändert hat.

Als Grundeinstellung wählen wir eine beliebige Szenerie und die Ansicht „Fester Beobachter“ (F5), das ist bei Panoramaszenerien die einzig verfügbare Ansicht. Ganz wichtig für das weitere Vorgehen ist es noch, mit der Tastenkombination <Strg - F12> die Physics Info zu aktivieren. Bilder mit aktivierter Physics Info sind im Verlauf dieses Artikels schon häufiger verwendet worden und sollten uns insofern vertraut sein. Beim normalen Fliegen ist diese Art der Darstellung eher störend, für das Tutorial ist sie jedoch elementar. Zur Auffrischung und Vervollständigung hier eine Liste mit den Bedeutungen der Farben und Formen der Physics Info.

Kreuzlinien in weiß:	Ursprung des Modells, Schnittpunkt = Null
Quader in blau:	rigidbody
Linien in rot:	jointlinear, (Linienursprung = Position)
Ring in cyan:	wheelhull, propellerdisc
Körper in magenta:	aerofuselage (Außenhaut)
Linien in gelb:	Schnittlinien in Flügelflächen und Rudern

Achtung: Die gelben Linien zeigen nicht die im aerofuselage Objekt angegebenen Schnitte, sondern vom Simulator auf Basis dieser Daten errechnete Linien. In der Regel sind das acht Linien pro Fläche.

Bei allen Lektionen ist es sinnvoll, erst den Text in den Modellinformationen zu lesen, denn zumindest bei den ersten Lektionen sieht man, wenn man die Hinweise zur Bedienung nicht beachtet, sonst so gut wie nichts. Im Übrigen macht das Ganze sicher mehr Spaß, wenn man einigermaßen weiß, was gleich auf dem Bildschirm geschehen wird.

Lektionen 01 und 02

Die TMD Datei der Lektion 01 stellt mit ihren zwei Objekten telemetry und rigidbody das absolute Minimum einer TMD Datei dar. Das obligatorische telemetry Objekt dient dazu, die Quellen für die verschiedenen Ausgabewerte zu definieren. Mindestens der Wert für [Body] muss angegeben werden. Beim rigidbody können die Parameter [Inertia] und [C0] auch einfach weggelassen werden. [Inertia] wird vom Simulator aus [InertiaLength] berechnet, und [C0] gibt eine Verschiebung des Ursprungs an und ist daher in der Regel Null. Für viele Parameter gilt, dass der Simulator Standardwerte verwendet, wenn diese in der TMD nicht angegeben wurden. Daher sieht eine TMD Datei, die über den Modelleditor erzeugt wurde, auch immer voller aus als die selbst geschriebene oder als die Originaldateien von IPACS.

Dass die rigidbodies in den ersten beiden Lektionen einfach durch den Boden durchfallen, dürfte nach dem Studium dieses Artikels keinen mehr überraschen.

Lektionen 03 und 04

Speziell Lektion 03 stellt eine Ausnahme dar. Obwohl noch gar keine Graphik sichtbar ist, wird in den collisionhull Objekten wie zum Beispiel in [FuselageCollision] in der Zeile <[string8][Geometry][Fuselage]> bereits darauf referenziert. Das ist möglich, weil über die TGB oder TGC Datei im gleichen Verzeichnis dem Simulator die Optik des Modells bereits bekannt gemacht wird. Erst in Lektion 04 folgt erstmals die Anzeige dieser Optik auch für uns als Betrachter, und ab jetzt verzichten wir darauf natürlich auch nicht mehr. Mir war es nur wichtig, über diese ersten Schritte die bereits mehrfach beschriebene strikte Trennung von Physik und Optik auch einmal sichtbar zu machen.

Lektionen 05 und 06

Mit dem Objekt jointlinear und dessen Parametern haben wir ja bereits etwas Erfahrung. In diesen beiden Lektionen wird der praktische Umhang damit gezeigt. Die Joints sind zunächst eher zu locker. Damit kann man jedoch gut leben, denn auch beim virtuellen Modellbau gilt: Nach fest kommt ab! Joints, die mit zu hohen Werten belegt werden, lassen das Modell gleich beim Aufsetzen auf den Startpunkt auseinander brechen. Das kann man sich wie einen sehr spröden Körper vorstellen, sobald die geringste Erschütterung auf ihn einwirkt, zerbricht er.

Lektion 07

Interessant ist hier die Wirkung des Objektes wheelhull auf das Objekt rotatingrigidbody. Der aktuelle (Dreh-)Winkel des Rades wird über den jeweiligen (Dreh-)Winkel der wheelhull definiert:

```
<[uint32][AngleID][RightWheelHull.RotationAngle]>
```

Damit ist sichergestellt, dass sich das Rad nur dreht, wenn es Kontakt zum Boden hat. Genau wie im echten Modell, ist die Reibung am Boden der Antrieb für das Rad.

Lektionen 08 und 09

Die Funktionen der Fernsteuerung werden normalerweise über die Funkstrecke übertragen und im Modell vom Empfänger ausgewertet und dann den Aktuatoren (Servos, Schalter,...) zugeführt. Auch im Simulator ist das so, zumindest was die Auswertung und die Aktuatoren angeht. Im Gegensatz zum richtigen Modell, können im Simulator auch ohne Expander an einen Empfängerkanal mehrere Aktuatoren angeschlossen werden. Die Servos im Simulator sind deutlich vielseitiger als im Original. Sie können in einigen Parametern wie zum Beispiel Kurvenverlauf oder Stellgeschwindigkeit verändert und so der jeweiligen Aufgabe angepasst werden. Auch können mehrere Servos in Reihe geschaltet werden. Besonders interessant ist es, auf diese Art ein Einziehfahrwerk zu konstruieren. Ein Steuerservo macht aus dem Schaltwert (-100% oder +100%) einen Kurvenverlauf, der wiederum der Steuereingang für die Servos ist, die letztlich das Fahrwerk bedienen. Noch komplexer wird es, wenn das Fahrwerk im geschlossenen Zustand von Klappen abgedeckt wird. Dann müssen auch die Klappen synchron zum Fahrwerk bedient werden und es ergibt sich unter Umständen die Notwendigkeit, weitere Steuerservos zu installieren.

Lektionen 10, 11 und 12

Ohne Motor und Propeller geht bei der PT40 nichts. Die Lektion 11 ist wieder ein gutes Beispiel für die Trennung von Optik und Physik. Obwohl noch gar kein Propeller sichtbar ist, kann das Modell bereits bewegt werden. Erst mit Lektion 12 ändert sich das, jetzt stimmen Optik und Physik wieder.

Lektionen 13 und 14

Die Aerodynamik, die mit diesen beiden Lektionen ins Spiel kommt, ist elementar dafür verantwortlich, wie sich das Modell im Flug verhält. Die Rumpfschnitte werden in Höhe und Breite als Rechtecke oder Ovale von vorne nach hinten aus der 3D Zeichnung abgelesen. Immer da, wo sich im Querschnitt etwas Wesentliches ändert, wird ein Schnitt gesetzt. Wesentlich in diesem Sinn ist alles, was aerodynamisch Einfluss hat.

Bei den Flügeln wird immer da ein Schnitt gesetzt, wo sich am Aufbau etwas ändert. Das sind der Ursprung (am Rumpf) und die Flügelspitze, sowie alle Stellen, wo Ruder oder Klappen beginnen oder enden. Angegeben werden die aus der 3D Zeichnung ermittelten Werte für Flügelnase (TE / trailing edge), Flügelhinterkante (LE / leading edge) und natürlich der Abstand zur Mittelachse. Angegeben werden diese Schnitte immer mit steigenden Y-Achsen Werten. Der linke Flügel also vom Rumpf zur Flügelspitze und der rechte Flügel von der Flügelspitze zum Rumpf.

Lektionen 15 und 16

Jetzt geht es an die beweglichen Teile des Modells. Ganz offensichtlich haben bislang alle Ruder gefehlt, etwas subtiler war das Fehlen der Servogestänge. Letztere tragen jedoch maßgeblich dazu bei, das Simulator Modell echter aussehen zu lassen. Geflogen ist unsere PT40 ja auch schon ohne sichtbare Ruder, mit den Rudern und den Gestängen kommt so langsam Flugplatzfeeling auf.

Lektion 17

Das mit dem Flugplatzfeeling stimmt natürlich noch nicht ganz, denn eine PT40 mit Verbrennungsmotor sollte sich auch entsprechend anhören. Der Sound stellt in der TMD Datei neben der Physik und der Optik einen eigenen Bereich dar. Bei den meisten Modellen enthält dieser Bereich nur das Objekt soundengine, in dem für verschiedene Motordrehzahlen WAV Dateien als Soundquelle angegeben werden können. Weitere Parameter dienen dazu, dem Simulator mitzuteilen, bei welchen Drehzahlen und mit welchen Lautstärken zwischen diesen Soundquellen gewechselt werden soll. Die WAV Dateien geben in der Regel einen kurzen Abschnitt des Original Geräusches wieder. Der Simulator gibt eine Endlosschleife des Abschnittes wieder und moduliert die Tonhöhe. Bei der Erstellung von eigenen Sounddateien ist daher darauf zu achten, dass innerhalb der Aufnahmezeit keine Änderungen am Ton stattfinden. Sowohl Lautstärke als auch Tonhöhe sollten konstant sein und das Ende der Aufnahme muss nahtlos an den Anfang passen.

Lektion 18

Auch in dieser Lektion geht es nur darum, den Realitätsgrad zu erhöhen. Ein Verbrennungsmotor macht nicht nur ein typisches Geräusch, er hat normalerweise auch noch Abgasrauch. Über einige Parameter in den Objekten smoke und smokegraphics kann das Aussehen dieses Rauches nahezu beliebig eingestellt werden. Ich kann so entscheiden, ob mein Motor heute etwas fetter oder recht mager läuft. Alles also wie im richtigen Leben, lediglich an der Duftausgabe wird beim Hersteller noch gearbeitet ;-)

Lektion 19

Mit fortschreitendem Realitätsgrad werden die verbleibenden „Unschönheiten“ geringer. Vermutlich ist es kaum aufgefallen, dass die Ruder bisher im Falle einer Kollision optisch in den Boden eingetaucht sind. Mit dieser Lektion bekommen alle verbleibenden Teile ihre eigene collisionhull und „erkennen“ fortan den Widerstand des Bodens und aller anderen Crash-Objekte. Crash-Objekte sind Bäume, Häuser, einfach alles, woran ein Modell auch in der Natur anstoßen würde.

Lektion 20

Zum Abschluss gibt es eine kleine Quizaufgabe. Wer alle Ausführungen verfolgt und am Simulator nachvollzogen hat, sollte mit etwas Nachdenken diese Aufgabe lösen können. Nicht zu vergessen ist eine andere Vorgehensweise, die ich ganz am Anfang bei den Tipps und Tricks bereits erwähnt habe: Probieren. Diese Aufgabe kann natürlich auch durch probieren gelöst werden. Dabei wird automatisch das Wissen um die einzelnen Parameterwerte erhöht und es entsteht Sicherheit und Erfahrung im Umgang damit.

SCHLUSSWORT

Auch wenn dieser Artikel nun einige Seiten lang geworden ist, er enthält noch lange nicht alles, was es an Infos zu diesem Thema gibt. Vieles lässt sich auch gar nicht so einfach beschreiben und wird einem erst beim Umgang mit den Parametern und Dateien klar. Auf jeden Fall ausreichen sollte der Artikel, um mit dem virtuellen Modellbau anfangen zu können. Dass dies nicht ohne eine treibende Neugier und die Bereitschaft etwas Neues zu erlernen geht, ist denke ich selbstredend. Ich wünsche allen, die sich auf diesen Weg begeben, viel Spaß beim Erstellen der Modelle und natürlich auch beim späteren Fliegen damit.

Andreas Preissler