

LINKÖPINGS UNIVERSITET
Institutionen för Teknik och Naturvetenskap

TNM085 Modelleringsprojekt
Fluid Simulation

March 12, 2011

Gustav Lysell Smälänning, gusly514@student.liu.se
Markus Olsson, marol114@student.liu.se
Magnus Strinnholm, magst693@student.liu.se
Johan Åslund, johas103@student.liu.se

Handledare
Anna Lombardi, Ulf Sannemo

Abstract

Denna rapport beskriver skapandet och resultatet av ett simuleringsprojekt. Genom att ha tagit fram en fysikalisk modell för ett flygplan under en flygplansstart, för att därefter skapa simulerings- och animationsegenskaper utifrån fysiken, har vi i detta projekt producerat ett program för simulering av flygplansstarter.

I rapporten ges detaljer om hur fysikaliska uträkningar utförts i MATLAB, modellskapande i 3ds Max och animeringar i Java3D.

Resultatet kan inte ses som helt fysikaliskt korrekt, då vi från början introducerade förenklingar som motsäger ett sådant resultat. Detta för att avgränsa uppgiften och göra den möjlig för den tid som gavs. Målet var att skapa en simpel animering av en flygplansstart, som beror på fysik och av användare valda värden, vilket också uppnåddes i form av ett körbart Java3D-program.

Contents

1 Inledning	1
1.1 Bakgrund	1
1.2 Syfte	1
1.3 Metod och källor	1
1.4 Struktur	1
1.5 Avgränsningar och förenklingar . . .	1
2 Fysik	1
2.1 Fysikalisk modell	1
2.1.1 Formler	1
2.1.2 Boeing 777-300ER data . . .	2
2.1.3 Övrig data	2
2.2 Fysikaliska uträkningar	2
2.2.1 Eulers stegmetod	2
2.2.2 Simulering i MATLAB	2
3 Visualisering	3
3.1 3ds Max	3
3.2 Java3D	3
3.3 Implementering av fysik	3
3.4 Animation	4
4 Resultat	4
5 Diskussion	5
6 Referenser	6
7 Bilaga A: Förstorade bilder	7

List of Figures

1	Figur på flygplanmodellen med utritade krafter.	2
2	Plot av simuleringens resultat	3
3	3D-Modell som representerar ett flygplan av typen Boeing 777-300ER .	3
4	Gränssnittet innan användaren sätter igång simuleringen	5
5	Visualisering igång	5
6	Flygplanet lyfter	6

List of Tables

1	Boeing 777-300ER specifikationer . .	2
---	--------------------------------------	---

1 Inledning

1.1 Bakgrund

Denna rapport behandlar ett projekt i kursen TNM085 Modelleringsprojekt som hålls vid Linköpings Universitet.

1.2 Syfte

Målet och uppgiften för kursen var att skapa en grafisk simulering av ett fysikaliskt system.

Det system som valdes för detta projekt är en simulering av en flygplansstart. Boeing 777-300ER är det flygplan man utgått ifrån i de beräkningar, estimeringar och simuleringar som behandlats i projektet. Baserat på fysikaliska uträkningar ska flygplansstarten kunna simuleras med olika initialvärden, för att sedan visualisera hur de olika faktorerna påverkar flygplanets start och lyftning.

1.3 Metod och källor

Grunden i systemet är en fysikalisk modell som skapades för flygplansstarten, uppdelad i en del som behandlar flygplanet på marken och en annan variant för flygplanet i luften under stigningen. Dessa togs fram för att sedan verifieras av Ulf Sannemo och utvärderades i MATLAB.

Numeriska beräkningar i MATLAB, med Eulers stegmetod, gjorde det möjligt att plotta värden för flygplanets positioner i höjd i förhållande till tid under den simulerade flygplansstarten. Tester utfördes även i samma program för att kontrollera huruvida rimliga de värden hämtade för det aktuella flygplanet var.

Animation av simuleringen utvecklades i Java3D, med importering av modeller som skapats i 3ds Max. De beräkningar för simuleringen man kom fram till tidigare applicerades på ett objekt som representerade det flygplan som skulle utföra flygplansstarten. Flygplansobjektet placerades i en virtuell tredimensionell värld, men de beräkningar som påverkar flygplanet behandlar endast två dimensioner.

Kursens examinator har varit handledare och assisterade i olika delar av projektet under dess gång.

1.4 Struktur

Rapporten inleds med att beskriva problemet, förutsättningar och antaganden. Därefter presenteras framtagandet av den fysikaliska modellen, hur den simulerades i MATLAB och hur den implementerades med grafik och animation i Java3D. Rapporten avslutas med en presentation av resultatet följt av en kort diskussion.

1.5 Avgränsningar och förenklingar

För att uppnå ett bra resultat bör man göra simuleringen så realistisk som möjligt, men med tanke på projektets storlek, tillgång till information och projektmedlemmarnas kunskaper inom området gjordes en del avgränsningar och förenklingar.

Flygplanets lyftkoefficient och aerodynamiska motstånd beror på dess form. Information om detta finns inte tillgängligt för allmänheten och för att räkna ut dem krävs tester i lufttunnel. Detta var förstas inte möjligt att genomföra i projektet, därför sattes dessa till godtyckliga värden med upphov till rimliga resultat.

Flygplanet påverkas av vind och friktion mellan mark och hjul under sin uppstart. I detta projekt simuleras flygplansstarten med vindstillhet och utan påverkan av friktion. Vinden skulle förstås kunna göra stor påverkan, men togs bort för att förenkla simuleringen. Friktionen anses påverka minimalt, därav den förenklingen.

Vid en verklig flygplansstart påverkar flygplanets klaffar dess stigning. Eftersom fluider och aerodynamik inte beräknas i detta projekt gör inte klaffarna någon påverkan i simuleringen, istället ändras flygplanets vinkel automatiskt. Flygplanets vinkel påverkar förstas ändå de aktuella krafterna och stigningen.

2 Fysik

2.1 Fysikalisk modell

2.1.1 Formler

Ett flygplan som rör sig genom luften påverkas i huvudsak av fyra krafter [1], [2]. Dessa är drivkraft T , aerodynamisk motståndskraft D [4], lyftkraft L [3] och tyngd W .

ohh.png	Table 1: Boeing 777-300ER specifikationer			Anfalls 14
	Massa (<i>kg</i>)	Vingarea (<i>m</i> ²)	Lyfthastighet (<i>km/h</i>)	
	167000-300000	428	270-345	
	2.1.3 Övrig data			
	Luftdensitet vid 20 °C, $\rho = 1.2$ [9]			
	Lyftkoefficient, $Cl = 1.6$			
	Koefficient för aerodynamiskt motstånd, $Cd = 1.6$			
	2.2 Fysikaliska uträkningar			
Målet var att beräkna alla verkande krafter i varje tidssampel för att kunna bestämma flygplanets momentana acceleration, hastighet och position i såväl x- som y-led. Detta gjordes i olika steg. Först beräknades de resulterande krafterna i x- och y-led vid varje tidpunkt. Dessa gav med hjälp av Newtons andra lag (4) och känd massa de motsvarande accelerationerna.				

Figure 1: Figur på flygplanmodellen med utritade krafter.

$$F_{res} = ma \quad (4)$$

Newtons kraftekvation där F_{res} är resulterande kraften, m är massan och a är accelerationen.

$$D = \frac{\rho V^2 A C_d}{2} \quad (1)$$

$$L = \frac{\rho V^2 A C_l}{2} \quad (2)$$

$$W = mg \quad (3)$$

Där Cl är flygplansvingarnas lyftkraftskoefficient, Cd är flygplanskroppens aerodynamiska motståndskoefficient (som beror på flygplanets form och aerodynamik), A är vingarnas totala area, ρ är luftens densitet, m är massan och g är gravitationskraften.

2.1.2 Boeing 777-300ER data

Den flygplansmodell som valdes för simuleringen är Boeing 777-300ER. De data som behövdes om flygplanet i vår fysikaliska modell och som vi använt oss av presenteras i tabellen nedan.

[5], [6], [7], [8]

2.2.1 Eulers stegmetod

Den matematiska och fysikaliska modellen består av differentialekvationer. För att kunna beräkna hastigheter och positioner regelbundet under simuleringen måste därför en numerisk metod användas. I detta fall valdes Eulers enklaste stegmetod [10] eftersom den är lätt att implementera, är snabb att beräkna och ger ett tillräckligt bra resultat. Med accelerationerna som parametrar gav Eulers stegmetod hastigheter i x- och y-led i varje punkt. Dessa användes sedan i sin tur som parametrar i samma stegmetod för att ta fram positionerna uttryckta i x- och y-koordinater.

2.2.2 Simulering i MATLAB

För att verifiera modellen och undersöka rimligheten i resultaten användes MATLAB. Fysiken och Eulers stegmetod implementerades och simuleringens resultatet plottades i form av sträckan i y-led (som tydligt visar när planet lyft). Figur 1 nedan visar ett resultat av denna simulering (med, förutom standardvärden, massan 270 ton, 1020 kN maximal drivkraft och en hastighet på 290 km/h vid inledning av stigningen):

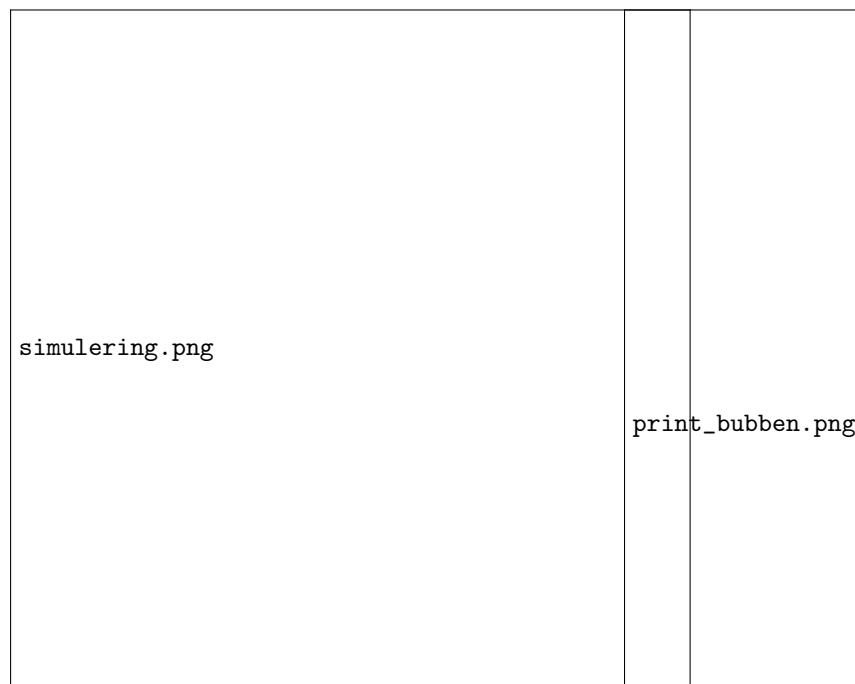


Figure 2: Plot av simuleringens resultat

Simuleringen visar att flygplanet inleder sin lyftning strax under 30 sekunder efter start. På figuren kan också tydas att flygplanets hastighet blir konstant 6.5 sekunder efter lyftning, eftersom att flygplanet då har roterat 14 grader och därefter är inställt på att behålla den då uppnådda stigningshastigheten.

3 Visualisering

3.1 3ds Max

För att gå från 2D till 3D användes först 3ds Max för att skapa 3D-modeller. Modellen av flygplanet utformades efter referensbilder på den faktiska flygplansmodellen. Förutom flygplanet skapades även en bakgrundsmodell, främst för att en bakgrund med varierande objekt förstärker uppfattningen av att planet rör sig, men också för att göra visualiseringen mer tilltalande. Modellerna exporterades i OBJ-format för att sedan kunna laddas in i Java3D för animering.

3.2 Java3D

Java3D är ett scengraf-baserat 3D-programmeringsgränssnitt för Java-plattformen

Figure 3: 3D-Modell som representerar ett flygplan av typen Boeing 777-300ER

som använder sig av OpenGL eller Direct3D. För att ladda in 3D-objekt från 3ds Max till Java3D krävdes användande av en så kallad "loader". Den gör det möjligt att importera OBJ-filer direkt från 3ds Max och göra dessa till fungerande 3D-objekt i Java3D. De fysikaliska uträkningarna i MATLAB behövde också överföras till körbar Javakod samt integreras tillsammans med animeringen av objekten i scenen. Ett grafiskt gränssnitt behövde även kopplas till de fysikaliska värdena som flygplanet berörs av för att man enkelt skulle kunna ändra på dessa. I Java3D används en trädstruktur av noder, så kallade "Transformgroups" som sköter translation, rotation och skalning i 3D-scenen. Dessa egenskaper ärvs uppåt i trädet.

3.3 Implementering av fysik

De fysikaliska formler, data och uträkningar som gjordes för MATLAB i kapitel 2 behövde implementeras i Java.

Alla värden som behandlades i uträkningarna skapades som variabler, i form av flyttal- och heltalstyp. De värden som beror på användarens inställningar eller andra värden sattes till 0 eller lämnades odefinierade för att inte påverka de fysikaliska uträkningarna från start.

De fysikaliska uträkningarna utfördes som i kapitel 2, koden översattes direkt från MATLAB-kod till Java-språk. Värdena som uträkningarna resulterar i, d.v.s. flygplanobjektets koordinatpositioner och hastighet i x- respektive y-led, läggs för varje tidssteg till i varsin vektor. Dessa kan sedan nås av den del i programkoden som utför animeringen. Fysiken sker alltså inte i realtid med visualiseringen, det är resultatet av de fysikaliska uträkningarna som animeras.

3.4 Animation

I Java3D användes klassen `PositionInterpolator` för att animera flygplanets rörelse i x- och y-led. `PositionInterpolator` utför en translation av det valda objektet från en startposition till en slutposition. I detta projekt utfördes denna typ av translation mellan varje värde ur vektorerna med data för koordinatpositioner (med lämplig skalning för att göra det visuellt realistiskt), som skapades i fysikdelen av programmet. Man använder sig även av en alphaklass, som anger tid före och tid för utförandet av animationen. I detta fall är tiden för utförandet av animationen anpassat till ett värde som passar det valda tidssteget i fysikuträkningarna, dock aningen avrundat.

Det normala vore att utföra positionsändringen i x-led på flygplansobjektet för att få det att åka framåt, detta innebär även att man behöver ha en kamera som rör sig tillsammans med flygplanet. För att göra denna uppgift lättare användes ett knep som ger samma resultat, fast med mindre behov av kod. Istället utfördes positionsändringen på bakgrunden, fast i motsatt riktning. Kameran behöver då inte röras, utan kan vara placerad statiskt med flygplanet centrerat i vyn. På detta sätt använder man sig alltså av en illusion av att flygplanet rör sig i sidled, men det är ändå resultaten av de ursprungliga fysikuträkningarna som man utgår ifrån. Det är alltså inga egentliga fel med resultatet som visas upp. Positionsändringen i y-led behandlade dock flygplanet, då det endast var själva flygplansstarten som skulle animeras och det inte ansågs finnas något behov av att följa flygplanet på väg upp i luften.

För rotation av flygplanet används en liknande lösning som nämnts ovan, en annan interpolatorklass kallad `RotationInterpolator`. Denna klass fungerar i princip på samma sätt, med start- och slutposition samt en alphaklass för tidsinformation för animationen. Värden för ändringen av vinkeln på flygplansobjektet hämtas dock inte ur någon av de vektorer som de fysikaliska uträkningarna resulterar i, istället sker den som en interpolation mellan två redan kända vinklar. Eftersom man i detta projekt utgår ifrån att flygplanet ska ändra sig från 0.0 till 9.0 grader på fyra sekunder, därefter från 9.0 till 14.0 grader under följande 2.5 sekunder, utför man denna animation så enkelt som möjligt. Vid det tillfälle då flygplanet försöker lyfta sätts helt enkelt animationen igång. Detta gör att animationen av rotationen ej är fysikalisk korrekt vid de tillfällen då flygplanet misslyckas att lyfta, men anses ändå vara en bra animation för att visa just misslyckade försök till lyftningar. Anledningen till att denna animation inte är närmare kopplad till fysiken grundar sig på de avgränsningar och mål man från början satte för projektet.

4 Resultat

Resultatet av programmet i Java3D blev ett simpelt grafiskt användargränssnitt, samt en animation som utförs då användaren väljer att starta den samma. Samtliga figurer nedan finns även i större versioner i Bilaga A.

Användaren kan här ange massa och den maximala drivkraft flygplanet når upp till, samt den hastighet vid vilken flygplanet ska försöka inleda stigningen. Genom att trycka på knappen `Take Off` startas sedan animationen, med fysikaliska uträkningar beroende på de värden användaren angivit.

Följande figurer är exempelbilder ur programmet vid körning, dels precis vid start samt då flygplanet precis inlett sin stigning. Här kan användaren inte längre påverka flygplanets startvärden, därför är dessa fält ej visade som aktiva.

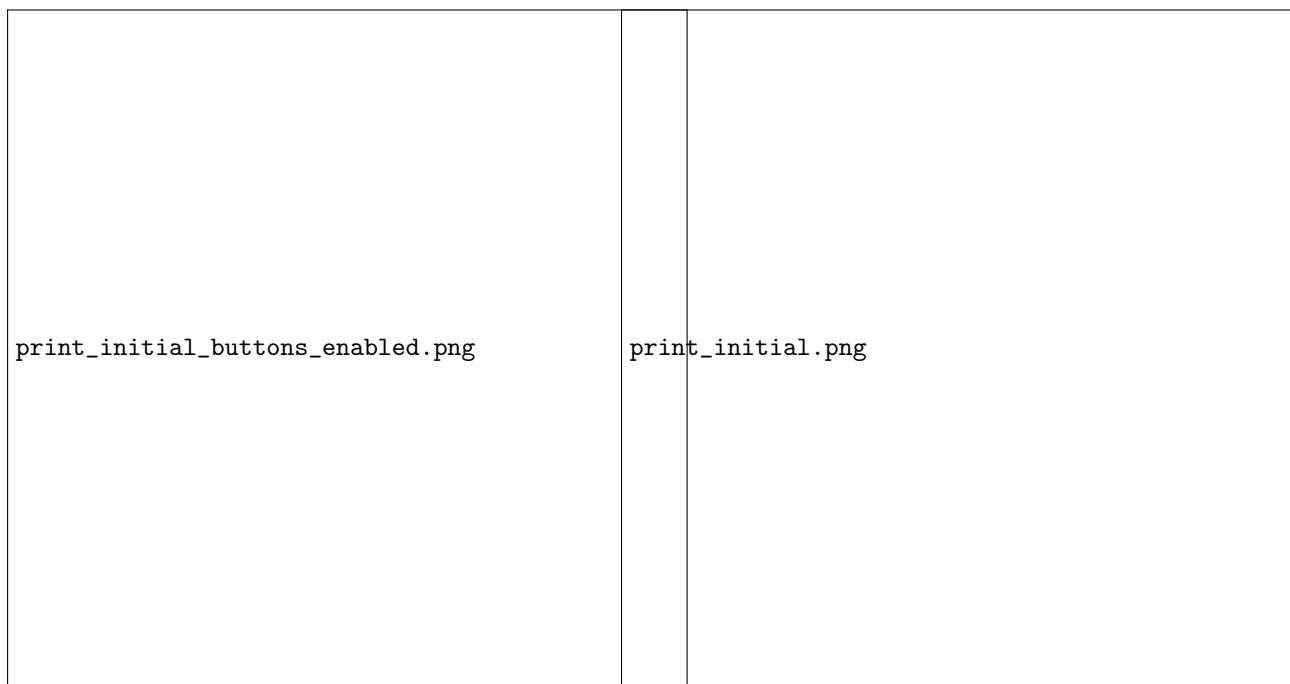


Figure 4: Gränssnittet innan användaren sätter igång simuleringen

Flygplanet är vid detta tillfälle i rörelse uppåt samt i rotation och bakgrunden är i rörelse åt höger, medan kameran står helt stilla.

A screenshot of a software interface, similar to Figure 4. It shows a large, empty rectangular area with a thin black border. In the bottom-left corner of this area, the text 'print_initial.png' is displayed in a monospaced font.

Figure 5: Visualisering igång

5 Diskussion

Arbetet har, överlag, fortgått effektivt, men en hel del problem dök också upp längs vägen. Vi var familjära med den grundläggande fysiken som gäller för flygplan i rörelse. Däremot var det svårt att komma fram till hur den skulle kunna beräknas i realtid och vilka storheter (acceleration, hastighet o.s.v.) som skulle vara viktigast att beräkna för att genomföra simuleringen på bästa sätt. Modellen gick trots detta ändå väldigt smidigt att ta fram. När den sedan skulle implementeras i MATLAB för evaluering så märkte vi att värdena till en början inte blev speciellt rimliga. Detta berodde främst på de osäkra konstanterna (C_l och C_d) som används i formlerna för lyft- och motståndskraft, eftersom dessa varit omöjliga att bestämma exakt. Efter en del experimentering med dessa så fick vi till slut ett någorlunda rimligt resultat på hastighet, sträcka innan lyft, lyfthastighet och stigningsgrad.

Nästa steg var då att skapa visualiseringen i Java, och detta gav oss många problem. Väldigt mycket tid har krävts just till Java3D-programmeringen. Det vi skulle vilja förbättra programmeringsmässigt, och även försökte med, är möjligheten att starta om simuleringen med en knapp för att slippa

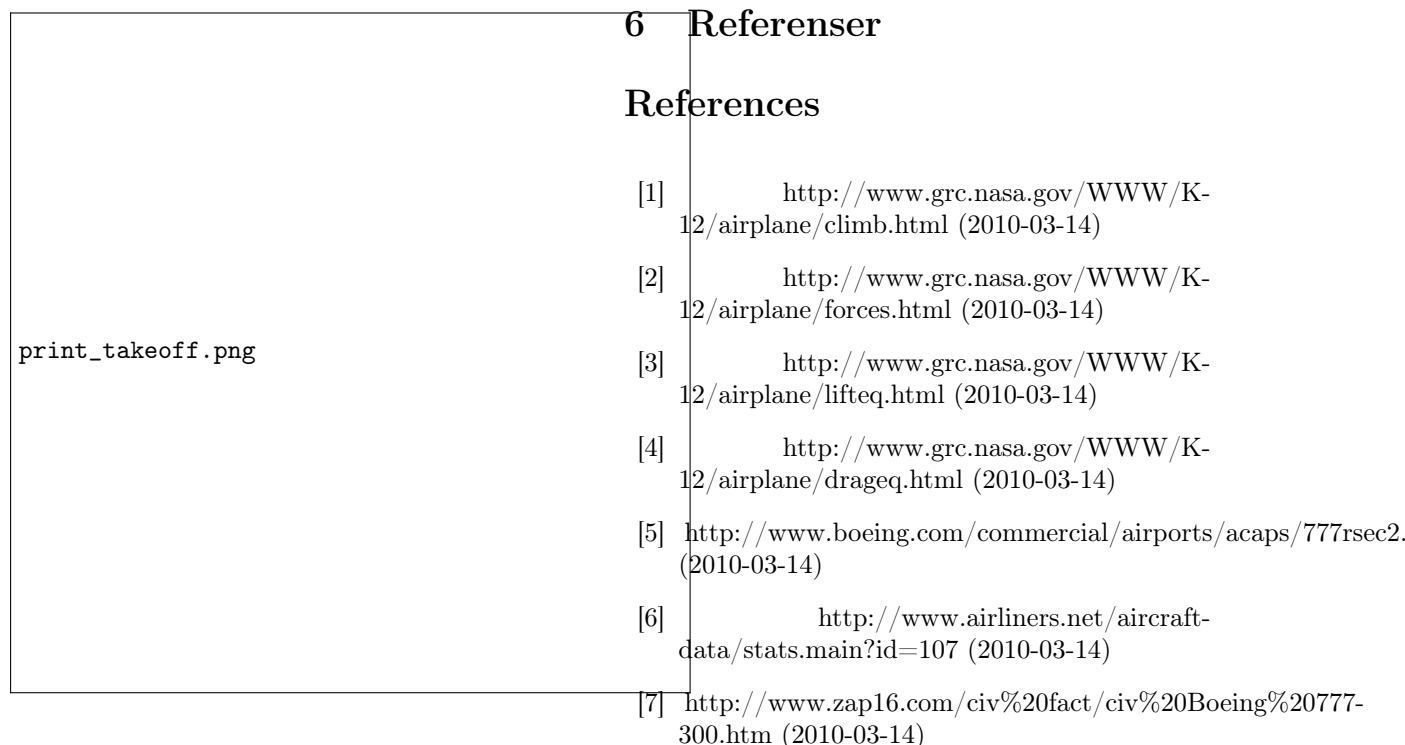


Figure 6: Flygplanet lyfter

köra om hela programmet vid varje körning, att visa värden på skärmen för exempelvis hastighet och sträcka under simuleringens gång och att skapa en körbar jar-fil för att kunna köra programmet på vilken PC som helst.

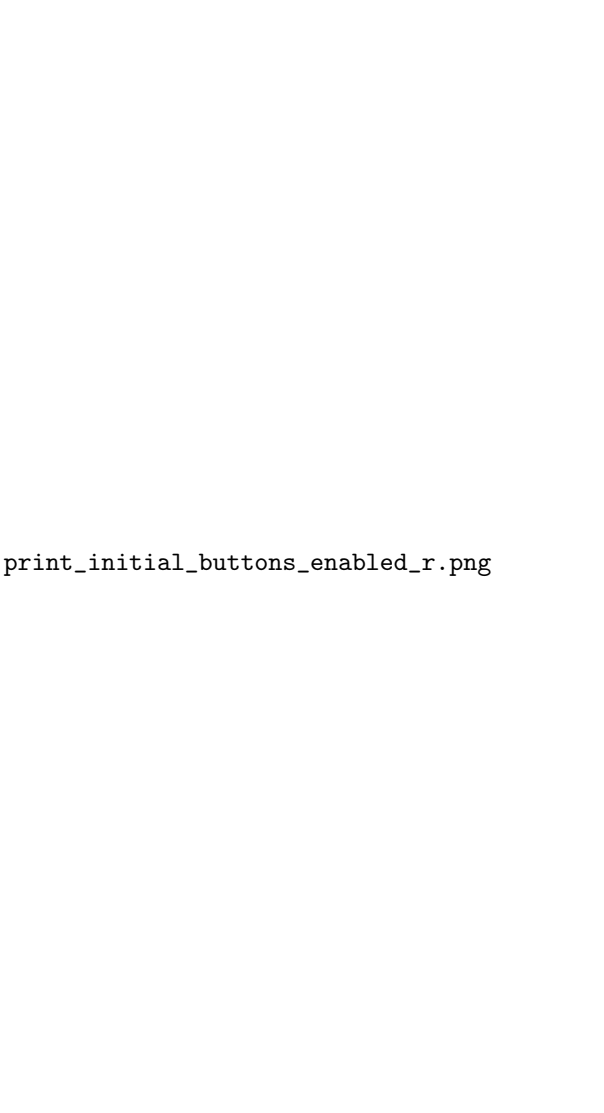
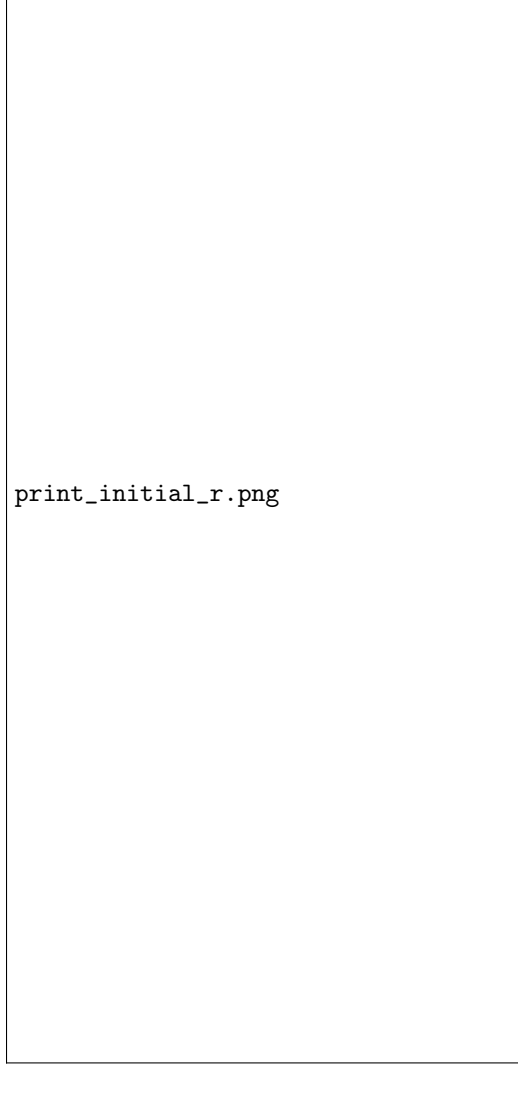
På det stora hela är vi nöjda med slutresultatet. Modellen är tillräckligt fysikaliskt korrekt för projektets mål och visualiseringen behandlar det vi hade planerat från början, nämligen en enkel flygplansstart för en Boeing 777-300ER.

6 Referenser

References

- [1] <http://www.grc.nasa.gov/WWW/K-12/airplane/climb.html> (2010-03-14)
- [2] <http://www.grc.nasa.gov/WWW/K-12/airplane/forces.html> (2010-03-14)
- [3] <http://www.grc.nasa.gov/WWW/K-12/airplane/lifteq.html> (2010-03-14)
- [4] <http://www.grc.nasa.gov/WWW/K-12/airplane/drageq.html> (2010-03-14)
- [5] <http://www.boeing.com/commercial/airports/acaps/777rsec2> (2010-03-14)
- [6] <http://www.airliners.net/aircraft-data/stats.main?id=107> (2010-03-14)
- [7] <http://www.zap16.com/civ%20fact/civ%20Boeing%20777-300.htm> (2010-03-14)
- [8] <http://www.bangaloreaviation.com/2009/05/typical-takeoff-and-climb-angles-of-all.html> (2010-03-14)
- [9] <http://sv.wikipedia.org/wiki/Ljudhastighet> (2010-03-14)
- [10] <http://www.sosmath.com/diffeq/first/numerical/numerical.ht> (2010-03-14)

7 Bilaga A: Förstoraade bilder

	
--	---

print_takeoff_r.png