

Treball de Fi de Grau

Grau en Enginyeria Informàtica

**Síntesi d'animacions amb físiques i
recuperació de l'equilibri**

25 de juny de 2020

Autor: Albert Ortiz Guillem

Directora: Nuria Pelechano Gomez

Especialitat: Computació

2019 - 2020 Q2

**UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH**

Facultat d'Informàtica de Barcelona



Abstract

This bachelor's thesis aims to study how to synthesize dynamic motion, for biped characters, in the field of real-time computer graphics. The solution proposed involves the development of different systems, that must work together, in order to produce animations that are aware of the surrounding environment. Particularly, a torque-based *ragdoll* controller, a full-body inverse kinematics solution and an equilibrium tracking algorithm. Using these systems, a complex biomechanical response such as equilibrium recovery is simulated.

Resumen

En este trabajo de fin de grado se lleva a cabo un estudio a cerca de la síntesis de animaciones dinámicas, para personajes bípedos, en el ámbito de gráficos por computador en tiempo real. La solución propuesta consiste en el desarrollo de distintos sistemas, que deben trabajar conjuntamente, a fin de producir animaciones consistentes con el entorno que les rodea. En concreto se tratará un sistema de control de *ragdolls* basado en fuerzas de torque, una solución de cinemática inversa para cuerpos humanos y un algoritmo para monitorizar el grado de equilibrio de un personaje. Usando estos sistemas, se simula una respuesta biomecánica compleja como es la recuperación del equilibrio.

Resum

En aquest treball de fi de grau s'estudia com sintetitzar animacions dinàmiques, per a personatges bípedes, dins l'àmbit dels gràfics per computador en temps real. La solució proposada consisteix a desenvolupar diferents sistemes, que han de treballar en sinergia, a fi de produir animacions conscients de l'entorn que les envolta. Concretament, un sistema controlador de *ragdolls* basat en forces de torque, una solució de cinemàtica inversa per a cossos humans i un algorisme per a monitorar el grau de desequilibri d'un personatge. Utilitzant aquests sistemes, se simula una resposta biomecànica complexa com és la recuperació de l'equilibri.

Índex

1	Introducció	6
1.1	Definicions importants	6
1.1.1	Animacions tradicionals i <i>procedurals</i>	6
1.1.2	Físiques de <i>ragdoll</i>	7
1.1.3	Cinemàtica inversa	8
1.2	El problema a resoldre	9
1.3	Actors implicats	10
2	Estat de l'art i justificació de l'alternativa de resolució escollida	11
3	Abast del projecte	14
3.1	Objectius genèrics	14
3.1.1	Subobjectius	14
3.1.2	Requeriments no funcionals	15
3.2	Obstacles i riscos	15
4	Metodologia i rigor	16
4.1	Desenvolupament iteratiu	16
4.2	Eines de desenvolupament	17
4.3	Mètode de validació	17
5	Planificació temporal	18
5.1	Desglossament de tasques	18
5.1.1	Gestió del projecte [T1-T8] i T19	18
5.1.2	Estudi algorismes d'animacions dinàmiques [T9]	19
5.1.3	Preparació de l'entorn de desenvolupament [T10]	19
5.1.4	Sistema de control de <i>ragdolls</i> [T11]	19
5.1.5	Sistema de cinemàtica inversa [T12]	19
5.1.6	Desenvolupament dels algorismes [T13-T16]	19
5.1.7	Resultats i defensa [T17-T18]	20
5.2	Recursos	20
5.3	Estimació temporal	20
5.4	Seqüència de desenvolupament: Gantt	23
5.5	Gestió del risc: Plans alternatius i obstacles	24
6	Pressupost	25
6.1	Identificació i estimació dels costos	25
6.1.1	Costos de personal i activitats	25
6.1.2	Costos genèrics	27
6.1.3	Costos associats al risc	28
6.1.4	Cost de contingència	28
6.1.5	Pressupost final	28
6.2	Control de gestió	29
7	Sostenibilitat	30
7.1	Autoavaluació	30
7.2	Dimensió econòmica	30
7.3	Dimensió ambiental	30

7.4 Dimensió social	30
8 Identificació de lleis i regulacions	32
9 Control de <i>ragdolls</i>	33
9.1 Definició del problema	33
9.2 Solució	33
9.3 Treball futur	36
10 Cinemàtica inversa	37
10.1 Definició del problema	37
10.2 Solució	38
10.2.1 Mètodes analítics	38
10.2.2 Mètodes numèrics	39
10.2.3 Mètodes basats en dades	40
10.2.4 Mètode emprat	40
10.3 Treball futur	47
11 Pèrdua d'equilibri	48
11.1 Definició del problema	48
11.2 Solució	48
11.3 Treball futur	51
12 Recuperació de l'equilibri	52
12.1 Definició del problema	52
12.2 Solució	53
12.2.1 Preliminars	53
12.2.2 <i>Ankle strategy</i>	54
12.2.3 <i>Hip strategy</i>	55
12.2.4 <i>Stepping strategy</i>	58
12.3 Treball futur	63
13 Conclusions finals	65
14 Bibliografia	66
Glossari	69
A Codi font	71

Índex de taules

1	Estimacions temporals per tasca.	21
2	Sou brut per hora treballada i anual.	25
3	Hores per tasca associada a cada treballador i cost final per tasca.	26
4	Costos genèrics.	27
5	Pressupost final.	28

Índex de figures

1	Evolució econòmica de la indústria dels videojocs	6
2	Exemple d'animació formada per quatre postures	7
3	Exemple de físiques de <i>ragdoll</i>	8
4	Exemple de cinemàtica inversa	9
5	Imatge derivada de l'animació (superior) i el <i>Movement Disgression Map</i> (abaix) que mostra els valors discrepants	12
6	Exemple de diferents transicions en funció de l'objecte amb el qual s'ha d'interactuar	13
7	Esquema de la metodologia <i>Agile</i>	16
8	Graf de dependències entre tasques.	22
9	Taula de Gantt	23
10	Rebut de la sol·licitud del permís de reutilització de les figures d'un article científic.	32
11	Escenari inicial	33
12	Resultat final	35
13	Els sis tipus d'articulació més comuns al cos humà	38
14	Esquema del <i>6R manipulator</i>	39
15	Aproximació lineal utilitzant la <i>Jacobian matrix</i>	39
16	Iteració completa de <i>FABRIK</i>	42
17	Com re-posicionar una articulació per complir restriccions de tipus <i>Ball-and-socket</i> (adalt) i <i>Hinge</i> (abaix)	44
18	Transformació del problema de la cinemàtica inversa, per a una cadena de dos ossos, a un problema trigonomètric bidimensional.	46
19	Resultat de l'algorisme de cinemàtica inversa. El personatge és capaç d'adoptar dinàmicament la postura que millor li permeti arribar als objectius (esferes verdes).	47
20	Projecció de K i s_i	50
21	Classificació dels segments	50
22	Tipus d'estratègies motores per a la recuperació de l'equilibri. (A,B,C) Mostren les <i>Ankle, Hip, Stepping strategies</i> respectivament, des d'un punt de vista lateral. (D,E,F) Mostren les mateixes estratègies frontalment.	52
23	L' <i>ankle strategy</i> en acció. El cub aplica una força sobre la part posterior del tors que es veu compensada amb l'aparent força exercida pels músculs dels turmells.	54
24	Esquema de l'IP	55
25	Esquema del mètode proposat per a simular una <i>hip strategy</i>	56
26	<i>Hip strategy</i> al rebre un impacte en diferents direccions	57
27	(a) Corba de Bézier de grau 3 definida pels punts P_0, P_1, P_2 i P_3 , (b) els polinomis base que la formen $B_{3,0}$ (blau), $B_{3,1}$ (verd), $B_{3,2}$ (vermell) i $B_{3,3}$ (cian)	59
28	Les diferents fases d'una passa (a) i la trajectòria típica (b) que s'ha utilitzat com a referència per a l'aproximació.	59
29	Trajectòria del peu aproximada amb una corba de Bézier.	60
30	Seqüència de la <i>stepping strategy</i> simulada.	63
31	Gràfic del rendiment de l'algorisme de recuperació de l'equilibri.	64

1 Introducció

La indústria dels videojocs ha patit un ràpid creixement des del seu origen. Aquest creixement ha sigut degut, en gran part, al progrés tecnològic de les últimes dècades. Gràcies a aquest progrés s'ha fet possible la creació de tota mena de plataformes, permetent una ràpida expansió del seu públic. Addicionalment, s'ha anat millorant el grau de realisme i interacció ofert als usuaris[1]. En la Figura 1 es pot apreciar amb més detall aquesta evolució en termes d'ingressos econòmics, així com l'efecte que ha tingut la incorporació d'aquests nous medis.

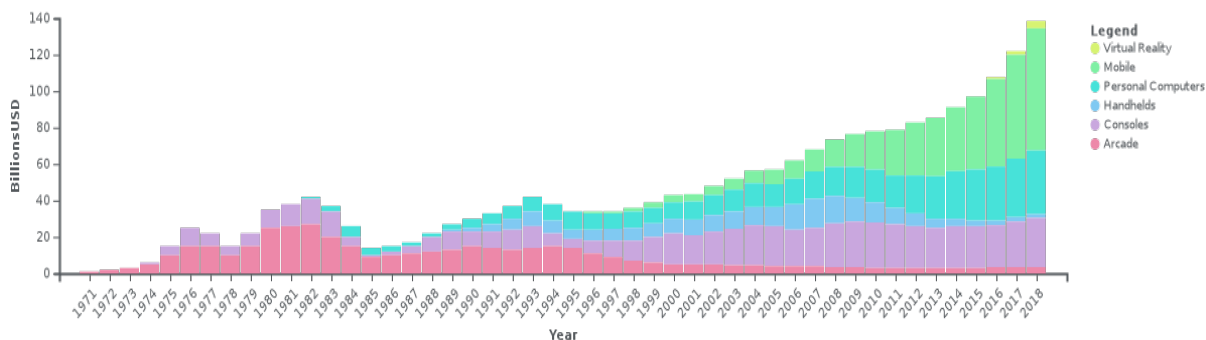


Figura 1: Evolució econòmica de la indústria dels videojocs

Font: <http://bloomberg.com/>

A causa del progrés mencionat anteriorment, també han anat augmentant les expectatives del seu públic. A mesura que la fidelitat visual és més elevada, s'espera també que els altres elements disposin d'una major complexitat. Això suposa nous reptes per als desenvolupadors. Un d'aquests reptes és la síntesi d'animacions per als personatges virtuals; camp en el qual la informàtica té una implicació clau.

Les animacions dels personatges són un dels principals vincles entre el videojoc i l'usuari. Per a cada esdeveniment que té lloc a l'entorn virtual, s'espera que hi hagi una animació coherent, en resposta, per part dels personatges afectats. Aquesta relació de causalitat té un gran impacte en l'experiència de l'usuari i, per tant, en la posterior recepció del producte final.

1.1 Definicions importants

Abans de detallar amb més exactitud quin és el problema que es vol tractar en aquest TFG, convé definir alguns conceptes d'especial rellevància:

1.1.1 Animacions tradicionals i *procedurals*

Les animacions, en el camp de l'animació digital, són un conjunt de diverses postures d'un personatge, disposades en un ordre concret, sobre una línia temporal. La interpolació d'aquestes postures, en funció del temps i de forma seqüencial, és el que acaba generant la il·lusió d'un moviment complex[2]. En la Figura 2 es mostra un exemple d'animació; el personatge té 4 postures repartides en els fotogrames: 1, 9, 13 i 17 respectivament. Les corbes verdes i vermelles representen com actuaria la interpolació sobre la posició de la pelvis i el cap del personatge, respectivament.

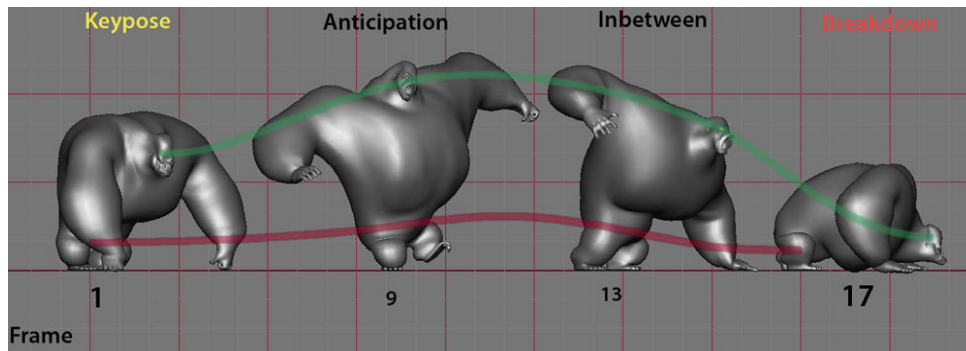


Figura 2: Exemple d'animació formada per quatre postures

Font: <http://robert-kuczera.com/>

L'animador sempre ha estat la persona encarregada de crear manualment aquestes animacions; sigui del no rés o a partir de dades extretes per captura de moviment. Aquest tipus d'animacions, les generades a partir d'un animador, seran referides com animacions tradicionals.

Amb els avenços de la informàtica, però, ha acabat essent possible generar animacions a partir d'algorismes. Aquestes animacions, generades de forma algorísmica, són el que es coneix com a animacions *procedurals*. Per tant, en resum, les animacions *procedurals*, a diferència de les tradicionals, no requereixen necessàriament la intervenció manual d'una persona per a la seva elaboració.

1.1.2 Físiques de *ragdoll*

Els motors de físiques, dins de l'àmbit dels videojocs, són programes informàtics que intenten realitzar una simulació de les físiques *Newtonianes*. Les simulacions solen ser bastant aproximades i sempre fent ús de valors discrets. Un cas d'ús d'aquests programes són les físiques de *ragdoll*. Aquestes físiques tenen com a objectiu principal simular el comportament físic d'un sistema d'objectes articulats com, per exemple, un cos humà. Quan un personatge es troba sota aquestes simulacions, se li sol anomenar directament *ragdoll*.

Cal tenir en compte que els *ragdolls* es troben formats per una jerarquia d'objectes físics simples com càpsules o ortodres. Entre aquests cossos existeixen vincles que els forcen a estar units entre ells i a complir amb límits rotacionals relatius.

A la Figura 3 es mostren diversos personatges on se'ls hi ha aplicat aquest tipus de simulacions. Com es pot veure, els seus cossos reaccionen de forma convincent envers el seu entorn. Gràcies a això, resulten especialment útils com a base per als sistemes que es volen desenvolupar en aquest projecte.

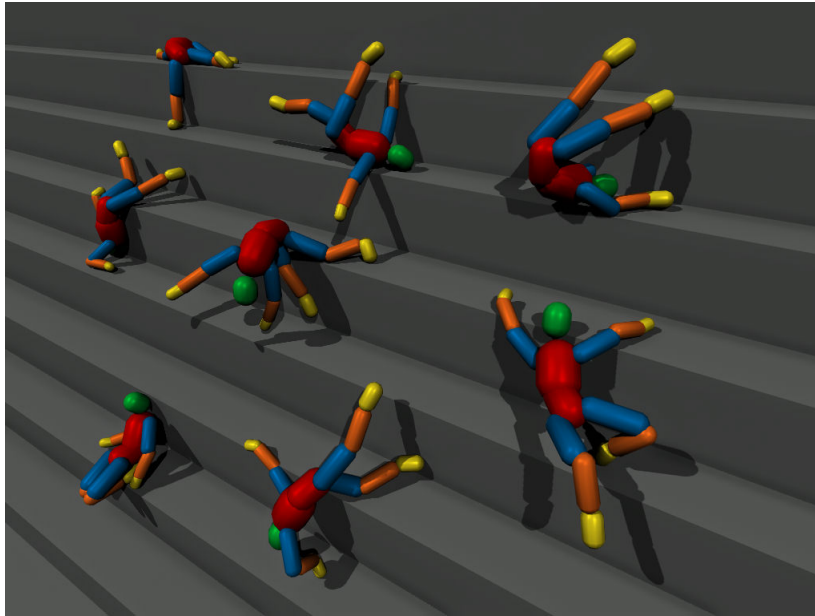


Figura 3: Exemple de físiques de *ragdoll*

Font: <http://developer.unigine.com/>

1.1.3 Cinemàtica inversa

La cinemàtica inversa és un procés matemàtic amb el qual, donat un sistema articulat¹ i una posició objectiu, s'obté una configuració del sistema on l'*end-effector*² es troba a la posició objectiu. Originàriament, el camp que més va incentivar l'ús d'aquesta eina va ser la robòtica. Amb el temps, però, camps com l'animació per computador també han estat molt beneficiats [3]. La Figura 4 mostra un exemple amb un sistema articulat format per 4 articulacions i una posició objectiu T. L'esquema superior mostra el sistema amb una configuració inicial. L'esquema inferior mostra al mateix sistema amb la configuració resultant després d'haver aplicat cinemàtica inversa.

¹Com un braç, per exemple

²Nom que rep en la literatura la part del sistema articulat que ha d'assolir la posició objectiu

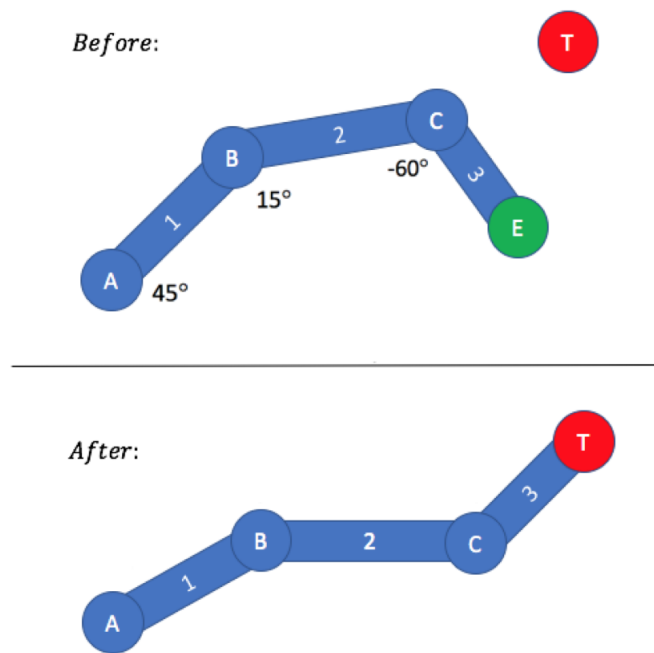


Figura 4: Exemple de cinemàtica inversa

Font: <http://gamasutra.com/>

1.2 El problema a resoldre

Actualment les interaccions físiques amb els personatges de la majoria de títols no solen ser gaire convincents, ja que, les animacions resultants, no acostumen a adaptar-se de forma fidel als esdeveniments externs. Per tant, el grau d'interacció i dinamisme que poden oferir és bastant limitat.

Generar animacions tradicionals és un procés car, tediós i estàtic³. Malgrat això, gràcies a la captura de moviment i que l'animador té un control total del resultat final, aquestes animacions solen ser molt realistes. Tot i això, per cada possible situació que hagi d'expressar un personatge, cal una animació específica[4]. Això comporta que sigui inviable produir aquest tipus d'animacions per cobrir esdeveniments complexos.

Utilitzar animacions *procedurals* des de zero, tampoc resol el problema satisfactòriament. És cert que, en tractar-se d'una simulació, els resultats no sempre són els mateixos i això ofereix dinamisme. També cal tenir en compte que, en tractar-se d'animacions generades en temps real, és relativament fàcil parametritzar-les a fi que s'adaptin a tota classe de situacions. D'altra banda, però, els moviments resultants solen ser poc fidels a la realitat.

Aquest TFG busca trobar un sistema híbrid que sigui capaç de combinar la fidelitat de les animacions tradicionals amb la flexibilitat de les animacions *procedurals*. Finalment, s'estudiarà

³Estàtic en el sentit que, un cop finalitzar el producte final, les animacions seran sempre les mateixes. L'usuari en molts casos acaba percebent aquesta regularitat, afectant en última instància a la qualitat del producte final

com utilitzar aquest sistema per implementar moviments més complexos com l'acte reflex de recuperar l'equilibri.

1.3 Actors implicats

Com s'ha exposat a l'anterior apartat, generar animacions per la via tradicional és car. Per a equips de desenvolupament amb una gran capacitat adquisitiva, això no sol representar un gran problema; inclús poden centrar-se a potenciar altres aspectes de les seves produccions a fi de compensar aquestes mancances. Amb un sistema com aquest, però, podrien augmentar la qualitat i flexibilitat dels seus productes amb un cost raonable.

En el cas del sector *indie*⁴, els equips solen rondar les 5 persones; amb perfils polivalents i una capacitat adquisitiva molt reduïda. Generalment, per això no es poden permetre implementar sistemes com el que es vol estudiar en aquest projecte. A part de millorar notablement els seus productes, tindrien la possibilitat d'explorar nous conceptes de *gameplay*, més arriscats i que les grans companyies difícilment explorarien.

Fora de l'àmbit dels videojocs, el sistema de balanceig potser podria ser adaptat en altres camps com la robòtica. El control directe de *ragdolls* també podria tenir un bon ús dins el camp de la realitat virtual.

Per últim, el Centre d'Investigació de Visualització, Realitat Virtual i Interacció Gràfica (ViRVIG)[5] potser pot treure partit d'algun dels algorismes desenvolupats.

⁴Sector format per equips amb un escàs o inexistent finançament

2 Estat de l'art i justificació de l'alternativa de resolució escollida

L'objectiu d'aquest projecte, com ja s'ha mencionat anteriorment, és crear un sistema que generi animacions dinàmiques. Es vol utilitzar com a base del sistema les físiques de tipus *ragdoll* per a obtenir uns moviments de base totalment adaptables a les diferents forces externes de l'entorn. El sistema estarà format per diversos sistemes, cadascun centrat en una tasca diferent. Per exemple, un mòdul es dedicarà a controlar les extremitats del *ragdoll* per a imitar una determinada trajectòria mentre que, un altre, s'ocuparà de generar aquestes trajectòries. La combinació d'aquests sistemes, serà el que en última instància permeti un sistema flexible i realista.

Anteriorment ja s'ha justificat per què aquesta solució és millor a l'alternativa d'utilitzar exclusivament animacions *procedurals* o tradicionals. El fet que se segueixi una filosofia de disseny modular, a més a més, permetrà que el sistema resultant sigui fàcil d'escalar i d'adaptar a tota mena de necessitats.

Les animacions dinàmiques es poden generar utilitzant diversos mètodes. Un dels més estesos a la literatura científica és el basat en dades. La solució consisteix a generalitzar les animacions utilitzant un algorisme d'aprenentatge automàtic [6] [7]. S'ha decidit no seguir aquesta línia perquè un dels objectius d'aquest TFG és que els algorismes siguin fàcils d'adaptar i reproduir. El problema dels mètodes basats en dades és que el resultat final depèn totalment de la quantitat i qualitat de les mateixes dades. Això en molts casos sol ser un requisit molt restrictiu. Existeixen bases de dades de captura de moviment d'accés lliure. Tot i això, moltes d'aquestes bases de dades tenen animacions sense netejar⁵[8], el qual pot provocar que els errors en les dades es propaguin als models resultants de l'aprenentatge automàtic.

Un altre mètode amb el qual es pot generar animacions dinàmiques és utilitzant algorismes de *reinforcement learning* (RL). Aquest mètode ha agafat bastant popularitat en els últims anys. De forma bàsica, aquests algorismes intenten que un agent aprengui a generalitzar les diferents situacions d'un entorn per a utilitzar un conjunt d'accions de manera que pugui maximitzar una determinada funció de guany[9]. Aquests algorismes es poden aplicar directament a un *radgoll* per a controlar directament el moviment articulari, mitjançant forces. L'inconvenient és que en molts casos aquests algorismes permeten als agents moure's de formes que poc tenen a veure amb el moviment real de les persones.

Actualment existeix un producte dins el mercat bastant famós anomenat *Euphoria* de l'empresa NaturalMotion[10]. Aquest producte se centra a oferir sistemes d'animacions dinàmics amb respostes complexes com les que es volen tractar en aquest projecte. Ofereix molts bons resultats i ha estat utilitzat en grans títols. El problema, però, és la seva inaccessibilitat per a petits estudis[11]. *Euphoria* és difícil d'integrar directament en un projecte i requereix que els mateixos enginyers de NaturalMotion modifiquin el codi base del client[12].

D'altra banda, existeix l'empresa RootMotion[13] que ofereix diverses eines per a la síntesi d'animacions dinàmiques. Les més destacables són *PuppetMaster* i *FINAL IK*. *PuppetMaster* ofereix un sistema per a realitzar transicions entre animacions tradicionals i físiques de *ragdoll*, *FINAL IK* ofereix un sistema de cinemàtica inversa. Aquests productes, tot i ser de pagament⁶[14],

⁵En la captura de moviment, com en qualsevol procés de digitalització, existeix soroll de diferent gravetat. Per aquest motiu, després de la captura de moviment, generalment cal un extens treball de neteja manual d'errors

⁶A dia d'avui, el preu total del producte és de 180\$

són considerablement molt més accessibles al públic general que Euphoria; els detalls de la seva implementació, però, estan ocults. En especial *FINAL IK* pot resultar molt difícil d'adaptar a un projecte; a causa del desconeixement del seu funcionament intern.

Per tant, actualment no existeix una solució al problema transparent i accessible a un públic majoritari. Aquest projecte, per tant, intenta suplir aquesta necessitat.

Dins l'àmbit acadèmic, la síntesi d'animacions ha sigut un tema molt estudiat durant les últimes tres dècades.

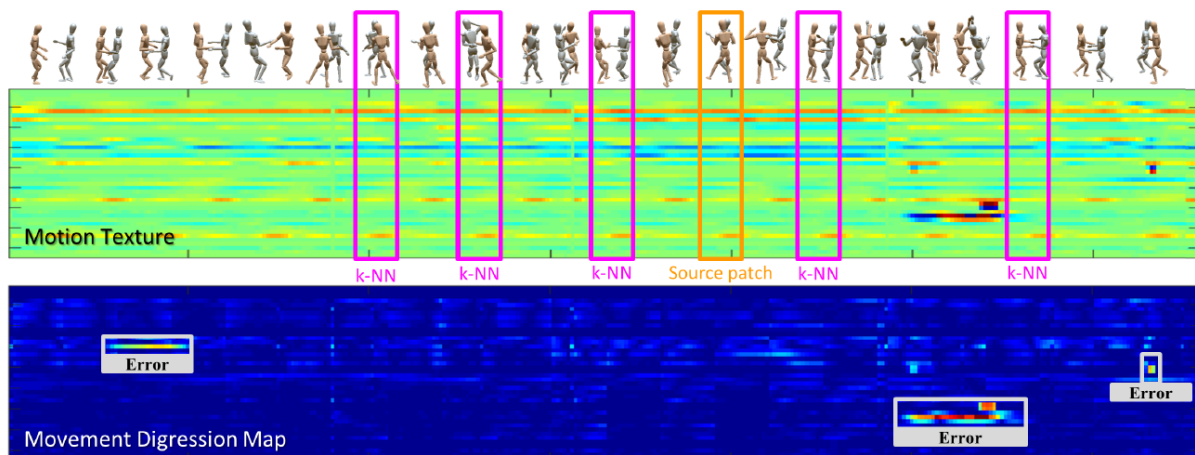


Figura 5: Imatge derivada de l'animació (superior) i el *Movement Digression Map* (abaix) que mostra els valors discrepants

Font: *Computer Graphics Forum*

En 2018 l'article *Self-similarity Analysis for Motion Capture Cleaning* [15] va presentar un mètode per a detectar de forma automàtica discrepàncies en animacions generades mitjançant captura de moviment. En la gran majoria de mètodes de captura de moviment, s'utilitzen uns marcadors situats en punts clau dels cossos dels actors. Mitjançant mètodes com infrarojos o càmeres, per exemple, es digitalitza la posició dels marcadors per a reconstruir postures en tres dimensions. El problema però, a part del soroll, és que l'oclusió temporal d'un o més marcadors pot generar grans discrepàncies en les postures resultants.

Generalment calen moltes hores, per part d'animadors professionals, per a detectar els problemes i polir-los. Amb la solució proposada en aquest article, utilitzant l'algorisme de *K Nearest Neighbors*, la descomposició d'una animació en forma d'imatge i un mètode novell per a comparar segments d'animacions⁷ de diferents longituds anomenat *dynamic time warping*; s'aconsegueix, no només detectar discrepàncies en una animació, sinó també un conjunt de possibles solucions per a cada postura errònia. La Figura 5 mostra la imatge generada a partir d'una animació, i la imatge generada a partir de l'anterior i una funció de similitud per a detectar errors.

L'any passat es va aconseguir desenvolupar un sistema que simula el moviment d'agents

⁷A l'article, aquests segments reben el nom de *Motion-words*

produït únicament amb sistemes musculars artificials, en temps real. El sistema és capaç d'incorporar personatges amb més de 300 músculs i, mitjançant deep learning, que aprenguin a utilitzar-los per a realitzar tota mena d'accions. De fet, el sistema va més enllà i és capaç de simular moviments amb agents que tenen deformitats congènites o han patit alguna intervenció mèdica[16].

També s'ha utilitzat recentment un algorisme de deep learning per aconseguir que personatges virtuals siguin capaços d'interactuar amb els objectes del seu entorn de forma natural. L'algorisme permet generar una seqüència de transicions que derivin en una transició suau, des de l'estat actual del personatge fins a l'acció final amb l'entorn. Aquesta acció final és capaç d'adaptar-se automàticament a qualsevol classe d'entorn. Per exemple, el personatge és capaç de seure en una gran varietat d'objectes sense importar la seva mida o forma. Anecdòticament, per a tal efecte, van necessitar una gran quantitat de dades de captura de moviment d'una persona assegant-se. Aquesta persona va haver de ser gravada seient en llocs de diferent forma i alçada i apropant-se des de diversos angles. El procés va portar centenars d'hores de rodatge. En la Figura 6 es pot apreciar les diferents transicions superposades derivades de diverses interaccions[17].

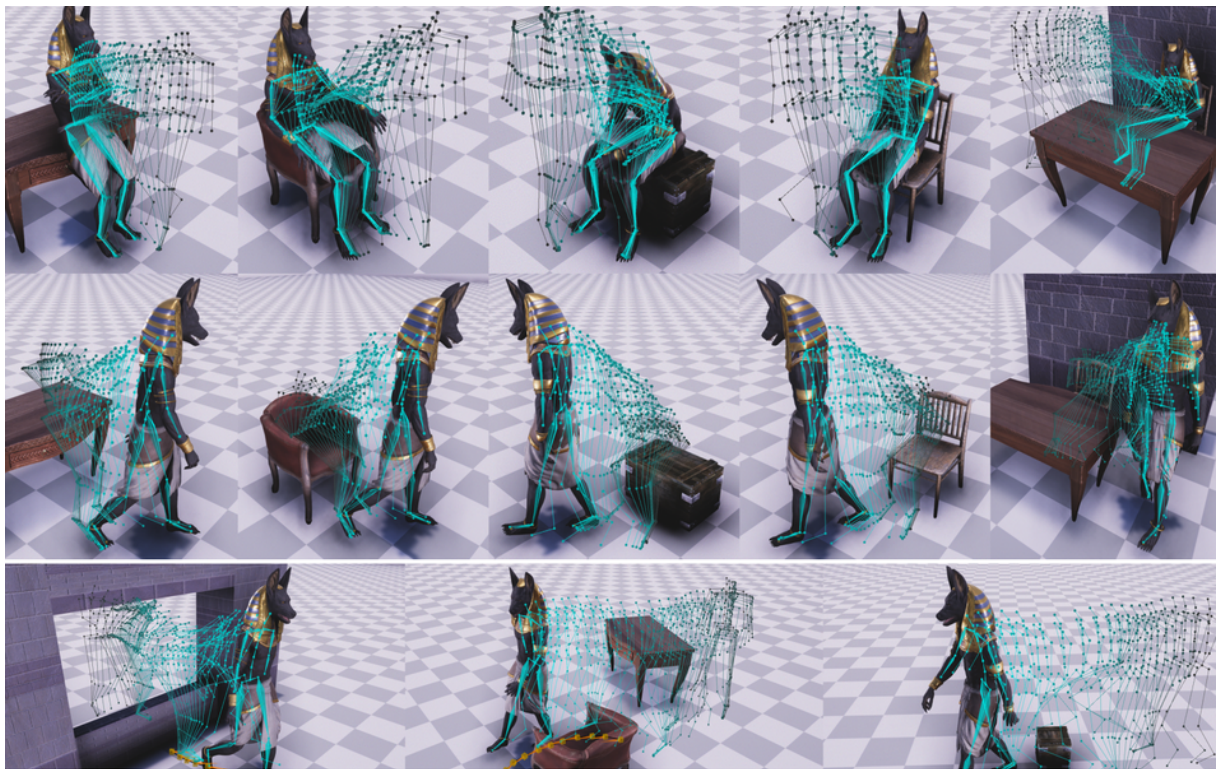


Figura 6: Exemple de diferents transicions en funció de l'objecte amb el qual s'ha d'interactuar

Font: Universitat d'Edimburg

Aquest any, l'article *Interactive character posing with efficient collision handling* [18] ha presentat un algorisme de cinemàtica inversa que incorpora restriccions capaces de respectar col·lisions i la conservació de relacions espacials.

3 Abast del projecte

Un cop contextualitzat el problema que es vol tractar, és necessari definir l'abast real que implica resoldre'l. En aquest apartat es definiran, tant els objectius genèrics, com els subobjectius corresponents. Per últim, s'analitzaran diversos obstacles i riscos que poden interferir en l'èxit del projecte.

3.1 Objectius genèrics

Com s'ha mencionat a l'apartat de contextualització, l'objectiu general d'aquest projecte és crear un sistema capaç de generar animacions dinàmiques, per a personatges digitals, que simulin respostes convinents envers forces físiques externes. Les respostes que es volen tractar, en ordre creixent de dificultat i preferència són:

- **Moviment conscient de l'entorn:** les articulacions dels personatges hauran de respondre de forma convincent en rebre qualsevol força de l'entorn mentre intenten imitar el moviment generat per un motion graph⁸. Per a tal efecte caldrà buscar un mètode que combini les físiques de *ragdoll* amb el moviment generat pel graf.
- **Recuperació de l'equilibri:** envers forces majors, els personatges han de poder perdre l'equilibri i intentar recuperar-lo de forma dinàmica.

3.1.1 Subobjectius

Derivat dels objectius anteriorment esmentats, sorgeix un seguit de subobjectius a tenir en compte:

- **Control del *ragdoll*:** els *ragdolls* són simples simulacions físiques. Cal una solució que permeti intervenir de forma directa en aquestes simulacions, a fi de generar un moviment articular propi.
- **Cinemàtica inversa:** un cop assolit l'anterior subobjectiu, s'ha d'implementar suport per a cinemàtica inversa per permetre un control total de les extremitats.
- **Detecció pèrdua de l'equilibri:** cal trobar un mètode que permeti detectar quan l'agent es troba o no en un estat de desequilibri.
- **Recuperació de l'equilibri:** s'ha d'estudiar i implementar una solució, viable en temps real, que permeti a un personatge en situació de desequilibri moure, de forma intel·ligent, les seves extremitats per a recuperar-lo de nou.

De forma opcional, si el temps ho permet, es vol estudiar una segona resposta complexa. Aquesta resposta consisteix a la d'autoconservació en cas de detectar un impacte imminent. Per tal d'implementar-la, caldrà crear dos sistemes: un de percepció de l'entorn per part de l'agent, on podrà detectar de forma activa possibles perills i un sistema que controli les extremitats del personatge a fi de protegir-lo envers aquests perills.

⁸Vegis el glossari per a una definició

3.1.2 Requeriments no funcionals

D'altra banda, es tindran en compte els següents requeriments no funcionals per al sistema final:

- Ha de ser flexible, modular i fàcil d'adaptar a noves necessitats.
- Ha d'oferir respostes convinents en un 80% dels casos, essent aquest 20% restant situacions extremes difícils de resoldre.
- Ha de ser eficient i fàcil d'integrar en qualsevol aplicació de gràfics per computador en temps real.

3.2 Obstacles i riscos

Un dels principals obstacles que s'haurà d'afrontar és la limitació de recursos computacionals. A causa de la naturalesa computacional dels videojocs, no es pot optar per qualsevol solució; el rendiment és un factor clau. És necessari trobar un bon compromís entre el cost computacional i la qualitat del resultat final. Per aquest motiu, simulacions realistes es troben fora l'abast del projecte i caldrà buscar mètodes aproximats. Si una solució no ofereix el rendiment desitjat, o bé es descartarà o bé s'intentarà simplificar. En cas que s'hagi de descartar, es dedicarà més temps a investigar possibles alternatives i/o abordar el problema des de noves perspectives.

Un altre obstacle a destacar és el fet de trobar un camí sense sortida en el desenvolupament dels algorismes. Els articles mencionats donen un bon indicatiu de com procedir però, potser, no s'acaben adaptant del tot a les necessitats del projecte. En cas de trobar aquest obstacle, es prendrà la mateixa mesura que amb l'anterior: invertir més recursos en la investigació de possibles alternatives.

També cal tenir en compte el risc que es pugui perdre dades per qualsevol mena de causa interna o externa al mateix desenvolupament. S'utilitzarà l'eina *GitHub* a fi de preveure aquest escenari. D'aquesta manera, sempre hi haurà una versió emmagatzemada al núvol des de la qual es podrà recuperar el projecte.

Per últim, destacar el risc de no poder complir els terminis. Segurament no es puguin explorar totes les alternatives disponibles i, per tant, caldrà una bona planificació. Durant el desenvolupament es realitzarà un control seguit dels terminis per detectar el compliment d'aquest risc. En cas necessari, s'utilitzaran tres setmanes addicionals, que s'han deixat lliures com a marge d'error, per intentar acabar el desenvolupament.

4 Metodologia i rigor

Com s'ha expressat en la secció anterior, un dels principals riscos que s'afrontarà en aquest projecte és el temps limitat. A part, l'existència de diversos subobjectius, alguns d'ells dependents d'altres, força encara més a determinar una correcta metodologia de treball.

En aquest apartat es justificarà quina metodologia ha sigut escollida i els motius de la decisió. Finalment, es descriuran les eines que es pretenen utilitzar per a desenvolupar el projecte i com es validaran els objectius.

4.1 Desenvolupament iteratiu

El mètode de desenvolupament escollit es troba basat en l'*Agile* [19]. La idea principal d'aquesta filosofia de desenvolupament consisteix a fraccionar el desenvolupament d'un gran projecte en iteracions o *sprints*. A la Figura 7 es pot apreciar gràficament com, per cada iteració, es realitzen les mateixes tasques que en un mètode cascada: planificació, disseny, desenvolupament, verificació, control i publicació. Aquestes iteracions tenen un objectiu de menor abast que el global, definit pel projecte, permetent que aquestes iteracions siguin flexibles i breus.

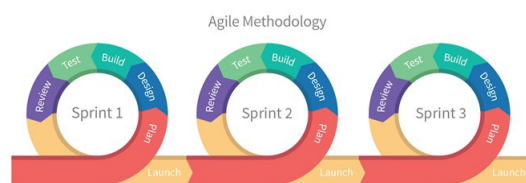


Figura 7: Esquema de la metodologia *Agile*

Font: <http://medium.com/>

El principal motiu pel qual s'ha decidit escollir aquesta metodologia és degut a la naturalesa modular d'aquest projecte, que permetrà una fàcil fracció en iteracions. En segon lloc, les iteracions permeten una detecció ràpida de possibles riscos i/o dificultats, oferint molta flexibilitat. Per acabar, un altre aspecte molt important d'aquesta metodologia és el fet de contar, des de les primeres iteracions, amb una primera versió sòlida. Això facilita, que si es falla en assolir un dels objectius, poder comptar amb un producte que compleix parcialment el desitjat.

Aquesta metodologia s'integrarà en el desenvolupament del projecte de la següent forma: s'intentarà establir una base mínima del projecte, que sigui viable assolir en un marge de temps relativament breu. Un cop assolida, s'aniran incorporant els diferents subobjectius, anteriorment mencionats, en terminis curts i tenint en compte les seves dependències. Durant el transcurs de cada iteració, es tindrà en compte que els canvis que s'incorporin siguin sòlids i funcionals. Al final de cada iteració, amb l'ajut de la directora del projecte, s'avaluarà l'èxit de la planificació inicial. Això permetrà establir eficaçment les millores de cara a la pròxima iteració.

A fi de supervisar l'èxit de les iteracions i, per tant, dels objectius, s'utilitzarà un indicador de compliment. Aquest indicador supervisarà que cada una de les tasques establertes en la planificació a priori de cada *sprint* es compleix.

4.2 Eines de desenvolupament

Tot el desenvolupament es durà a terme mitjançant el motor gràfic Unity3D, fent ús de la versió 2019.3.2f1. En un principi també es va plantejar utilitzar Unreal Engine. Finalment es va descartar perquè, d'entre altres motius, Unity3D incorpora eines d'aprenentatge automàtic que poden resultar crucials per assolir els últims objectius. No es pretén utilitzar cap llibreria externa per a desenvolupar els sistemes desitjats, únicament l'API de Unity3D⁹

GitHub s'utilitzarà per mantenir un seguiment de les diferents iteracions del projecte i poder recuperar una versió anterior; en cas de qualsevol mena d'incident.

Finalment, Trello s'utilitzarà per a gestionar els diferents objectius de cada iteració.

4.3 Mètode de validació

Gràcies a la metodologia de desenvolupament escollida, es podran programar reunions amb la directora del projecte per avaluar l'assoliment de cada un dels objectius actuals i discutir les pròximes millores.

⁹<https://docs.unity3d.com/ScriptReference/>

5 Planificació temporal

La càrrega acadèmica del TFG correspon a 18 crèdits. La normativa de la FIB indica que cada crèdit equival aproximadament a 30 hores[20]. Tenint en compte això, s'espera que la durada del projecte sigui de 540 hores. També cal tenir en compte que la majoria de projectes tenen moltes més tasques de les que a priori ens podem imaginar. Per tant, la finalitat d'aquesta secció és doble: identificar de forma realista quines són aquestes tasques i demostrar que la seva realització és factible en 540 hores. Per a demostrar-ho, s'utilitzarà una taula de Gantt juntament amb una estimació temporal i un pla d'acció.

L'inici del projecte va tenir lloc el 17/02/2020 i es preveu la seva finalització definitiva el 3/07/2020. S'intentarà treballar una mitja de 7 hores diàries de dilluns a diumenge.

En aquesta secció es descriuen totes les tasques que s'han considerat necessàries per a realitzar el TFG. També s'expliquen les diverses relacions de dependència. A cada tasca se li ha associat un codi per a facilitar la seva referència.

5.1 Desglossament de tasques

5.1.1 Gestió del projecte [T1-T8] i T19

Aquí es troben agrupades totes aquelles tasques relacionades amb la mateixa gestió del projecte. Primerament, la planificació general del projecte; per raons òbvies, necessita ser completada primer. Dins de la planificació general cal destacar: la definició de l'abast, la planificació temporal i la creació del pressupost i l'anàlisi de sostenibilitat. En segon lloc, les tasques relacionades amb la metodologia *Agile* que es vol dur a terme. Això inclou: una planificació inicial de totes les iteracions, les reunions de control, els tests de qualitat i les planificacions prèvies a cada iteració.

La planificació inicial de totes les iteracions té com a objectiu establir un nombre d'iteracions a priori i distribuir les diferents tasques entre elles. Aquesta distribució s'efectuarà tenint en compte les diferents relacions de dependència entre tasques i les diferents estimacions temporals en relació amb el temps disponible. Les reunions de control tindran com a objectiu supervisar l'èxit de cada una de les iteracions i realitzar possibles adaptacions en la planificació. Es realitzaran al final de cada iteració. El test de qualitat s'efectuarà en finalitzar la implementació d'una iteració i es basarà en verificar el correcte funcionament dels canvis introduïts. Per acabar, les planificacions prèvies tindran com a objectiu establir una gestió intel·ligent del temps disponible per a desenvolupar cada iteració i s'efectuarà a l'inici de cada una d'elles.

Per últim, en aquest apartat també s'incorpora la tasca de documentació, que es realitzarà de forma simultània amb qualsevol altra tasca de les mencionades en aquest document. Les hores de documentació, per tant, no es tindran en compte en el còmput total; ja que es troben de forma implícita a la resta de tasques.

En resum, les tasques a considerar dins d'aquest apartat són:

- **Abast [T1].**
- **Planificació [T2].**
- **Pressupost [T3].**

- **Sostenibilitat [T4].**
- **Agile: Planificació general [T5].**
- **Agile: Planificació prèvia [T6].**
- **Agile: Test de qualitat [T7].**
- **Agile: Reunió de control [T8].**
- **Documentació [T19].**

5.1.2 Estudi algorismes d'animacions dinàmiques [T9]

Com que aquest projecte té un gran component de recerca, per al desenvolupament d'alguns dels algorismes caldrà una bona inversió de temps en estudiar possibles solucions. Aquesta recerca es pot dur a terme paral·lelament amb el mateix desenvolupament iteratiu. En cas que una solució no s'adapti als objectius, el temps dedicat a aquesta tasca es veurà incrementat.

5.1.3 Preparació de l'entorn de desenvolupament [T10]

Aquesta secció especifica la tasca que es realitzarà abans de començar el desenvolupament. En particular, es crearà el projecte amb *Unity3D* i es recopilaran les animacions i models 3d necessaris.

5.1.4 Sistema de control de *ragdolls* [T11]

A fi de poder implementar els algorismes que generin animacions de forma dinàmica, caldrà tenir un control directe sobre el *ragdoll* dels personatges. En aquesta tasca es busca desenvolupar un sistema que cobreixi aquesta necessitat.

S'ha trobat diverses maneres de dur-ho a terme: des d'un control articulari directe i independent fins a la simulació d'un sistema muscular amb estímuls nerviosos. S'haurà estudiat quin és el sistema més adequat per aquest projecte; el qual pot incrementar el temps real de la tasca.

5.1.5 Sistema de cinemàtica inversa [T12]

Un cop finalitzat l'anterior sistema, es donarà suport a la cinemàtica inversa per a tenir un control més complex dels cossos. En particular es vol que, donades una articulació i una posició objectiu, el *ragdoll* sigui capaç de reorientar les seves articulacions fins a assolir la posició desitjada. Aquest sistema, conjuntament amb l'anterior, serviran com a base per al desenvolupament dels algorismes.

5.1.6 Desenvolupament dels algorismes [T13-T16]

Caldrà desenvolupar un seguit d'algorismes que facin possible: la detecció de pèrdua d'equilibri, la recuperació de l'equilibri, la detecció d'impactes imminents i una resposta d'auto-conservació. Tot i que aquests algorismes estan pensats per treballar de forma conjunta, poden ser desenvolupats en qualsevol ordre, ja que no hi ha cap mena de dependència tècnica, a part de les comunes, mencionades a l'anterior apartat.

L'algorisme que té major rellevància en aquest projecte, com ja s'ha comentat, és el de recuperació de l'equilibri. Donada la seva importància i complexitat, tindrà una major dedicació d'hores. L'algorisme de detecció d'impactes imminents, per tant, és secundari.

Tenint en compte això, les tasques són:

- Alg. detecció pèrdua equilibri [T13].
- Alg. recuperació de l'equilibri [T14].
- Alg. detecció impactes [T15].
- Alg. resposta impactes [T16].

5.1.7 Resultats i defensa [T17-T18]

Aquestes tasques són les últimes i consistiran a extreure les conclusions pertinents del projecte i preparar la defensa davant el tribunal.

Les tasques concretament són:

- Conclusió [T17].
- Defensa [T18].

5.2 Recursos

Un cop identificades totes les tasques, cal tenir en compte els recursos que faran falta per a dur-les a terme. La finalitat d'aquesta secció és descriure quins seran aquests recursos.

S'utilitzarà un ordinador de sobretaula amb un processador Intel Core i7-4790k (4.00GHz), 32GB de RAM, una GPU Nvidia GTX 970 amb 4GB de RAM i el sistema operatiu Windows 10. En l'àmbit de programari, com s'ha mencionat anteriorment, es farà ús del motor gràfic Unity3D amb la versió 2019.3.2f1 com a eina principal. També s'utilitzarà *LaTex* per a la documentació, el llenguatge de programació *Python* per a les eines d'aprenentatge automàtic i *Trello* per a la gestió de tasques.

Serà necessari un model 3d humà i animacions tradicionals de caminar, córrer i estar quiet.

Pel que fa a recursos humans, caldrà un cap de projecte que pugui dur a terme totes les tasques de gestió i un enginyer informàtic capaç d'afrontar els aspectes tècnics del projecte. Per últim, caldrà un reclutador que pugui seleccionar aquests perfils.

5.3 Estimació temporal

En aquesta secció es mostra la Taula 1 amb les tasques anteriorment mencionades i les hores que es preveu invertir així com les dependències entre tasques. La Figura 8 mostra de forma gràfica les dependències.

Tasca	Temps estimat en hores	Dependències
Abast[T1]	30	{}
Planificació[T2]	30	{T1}
Pressupost[T3]	30	{T2}
Sostenibilitat[T4]	8	{T3}
<i>Agile</i> : Planificació general[T5]	20	{T4}
<i>Agile</i> : Planificació prèvia[T6]	6	{}
<i>Agile</i> : Test de qualitat[T7]	20	{}
<i>Agile</i> : Reunió de control[T8]	6	{}
Estudi algorismes d'animacions dinàmiques[T9]	60	{}
Preparació entorn de desenvolupament[T10]	5	{T5}
Sistema control de <i>ragdolls</i> [T11]	50	{T10}
Sistema cinemàtica inversa[T12]	49	{T10}
Alg. detecció pèrdua equilibri[T13]	30	{T11, T12}
Alg. recuperació de l'equilibri[T14]	100	{T11, T12}
Alg. detecció impactes[T15]	30	{T11, T12}
Alg. resposta impactes[T16]	20	{T11, T12}
Conclusió[T17]	23	{[T13-T16], T19}
Defensa[T18]	23	T17
Documentació[T19]	70	{}
Total	540	[T1-T19]

Taula 1: Estimacions temporals per tasca.

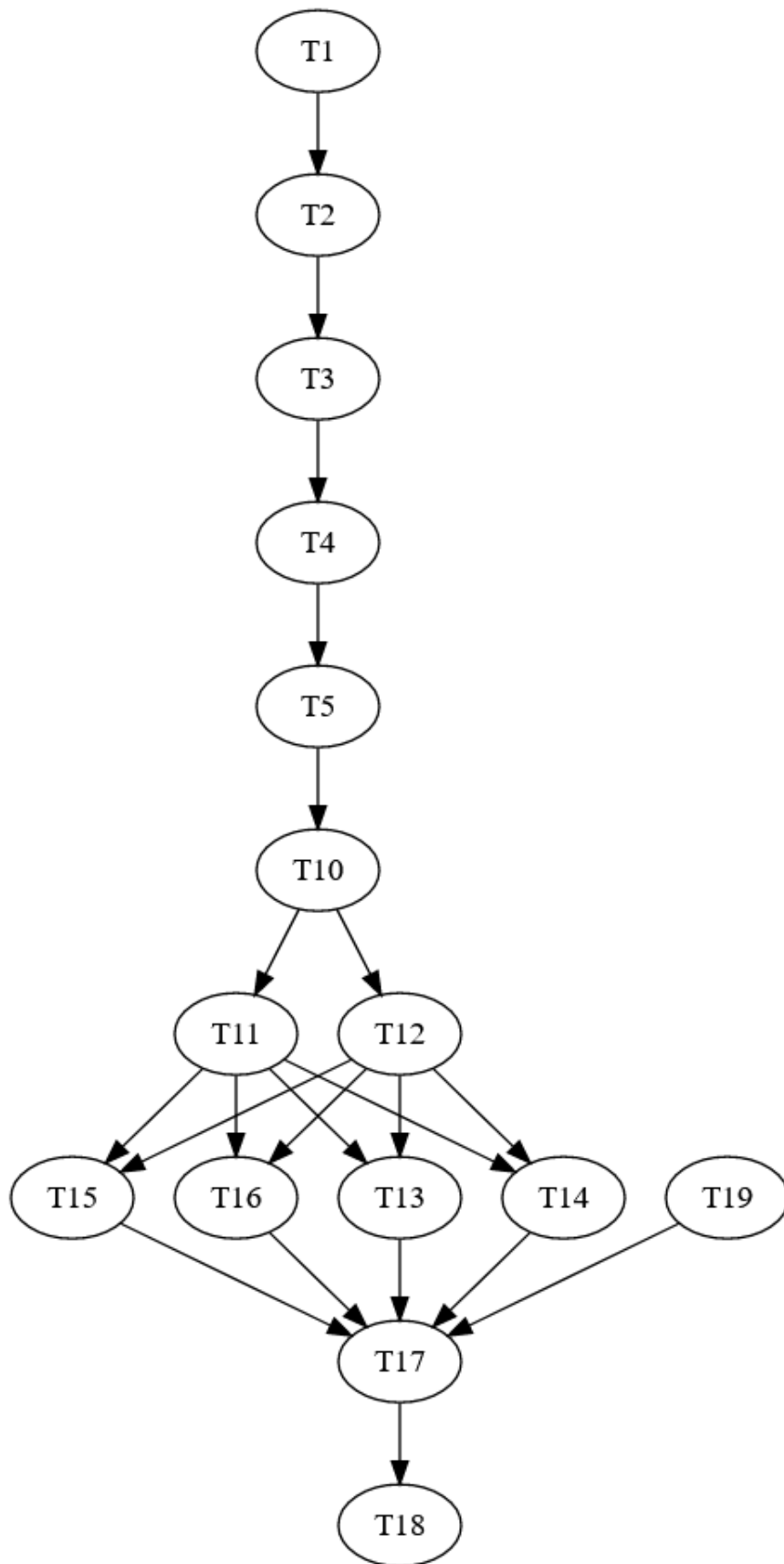


Figura 8: Graf de dependències entre tasques.

5.4 Seqüència de desenvolupament: Gantt

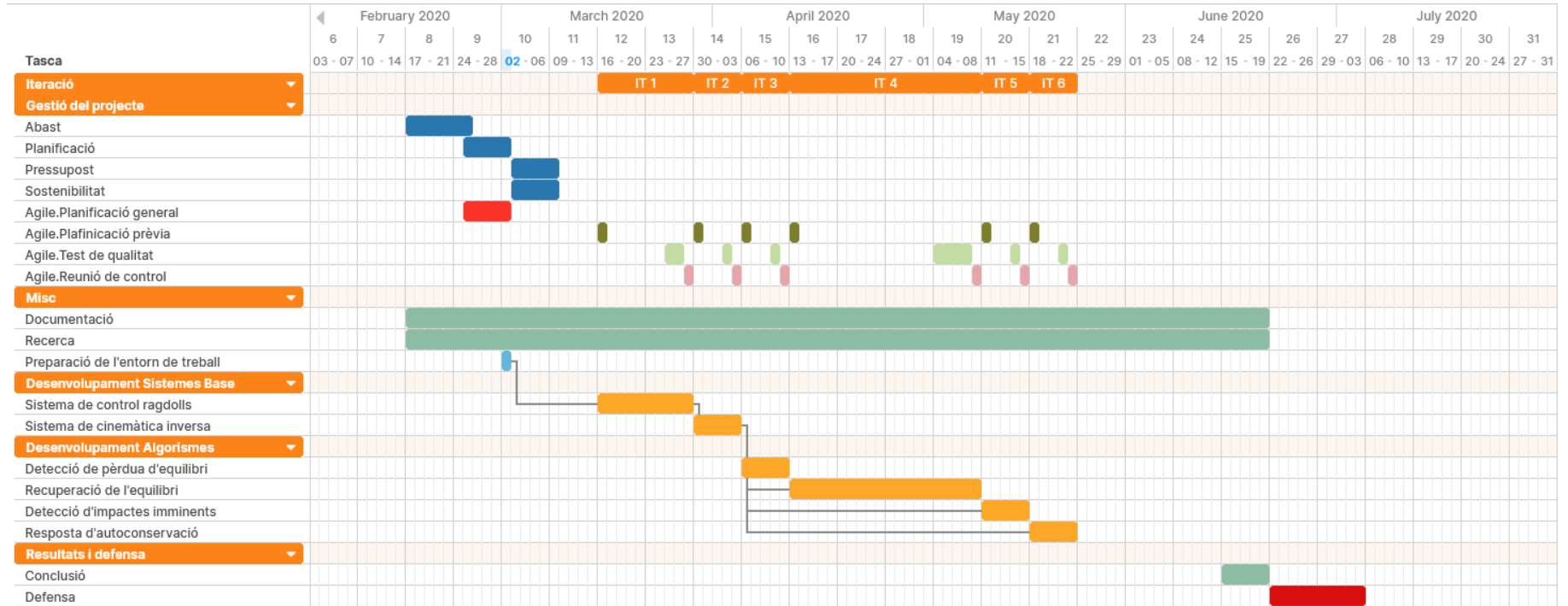


Figura 9: Taula de Gantt

5.5 Gestió del risc: Plans alternatius i obstacles

Com es pot apreciar a la Figura 9, la idea és realitzar un total de 6 iteracions; cadascuna dedicada a implementar un sistema en concret. També es pot observar que hi haurà reunions amb la directora entre les iteracions, a fi de controlar l'èxit dels diferents objectius. La tasca de documentació apareix constantment, ja que la idea és documentar de forma extensiva, inclòs durant el mateix desenvolupament. Els caps de setmana no apareixen en la taula per simplificar la lectura, però es tenen en compte en el còmput d'hores.

En referència a l'apartat 3.2, un dels principals obstacles que s'ha tingut en compte és el del temps. Per això, com que una de les tasques clau del projecte és l'algorisme de recuperació de l'equilibri, s'ha decidit sobreestimar-la per evitar a tota costa el seu incompliment. El sistema de control de *ragdolls* també ha sigut sobreestimat, en relació amb el risc que es va comentar de no saber quina solució en concret s'utilitzarà.

Un altra tasca que s'ha tingut en compte dur a terme, de forma extensiva, durant el decurs d'aquest TFG, és la de recerca. Això és degut al fet que existeix el risc de trobar un camí sense sortida desenvolupant els diferents algorismes i, per tant, caldrà investigar noves alternatives.

Per últim, s'han deixat tres setmanes lliures entre el final de la iteració 6 i la conclusió del projecte. Aquestes tres setmanes serveixen com a marge de maniobra, en cas que alguna de les iteracions fracassi o s'hagi de realitzar alguna rectificació en la planificació general. Aquest pla alternatiu suposaria, aproximadament, un total de 60 hores. En cas que aquestes tres setmanes no fossin suficients, s'evitaria fer la resposta secundària de prevenció d'impactes i es distribuïrien les seves hores estimades en aquelles tasques, més prioritàries, que encara estiguessin incompletes.

6 Pressupost

En aquest apartat s'analitzen quins són els factors econòmics implicats en la realització del projecte. S'estimen els preus en forma de pressupost i s'ofereix un pla d'acció en funció dels possibles riscos que puguin alterar aquesta estimació, a fi d'adaptar-la eficaçment.

6.1 Identificació i estimació dels costos

6.1.1 Costos de personal i activitats

Abans d'entrar en detall amb els costos d'activitats, cal tenir en compte quin personal hi estarà al càrrec; ja que el seu sou tindrà una relació directa amb el cost final de cada activitat.

En primer lloc, serà necessària una persona de recursos humans capaç de reclutar el personal requerit.

En segon lloc, caldrà un cap de projecte capaç de realitzar totes aquelles tasques relacionades amb la gestió i presentació del projecte.

En tercer lloc, caldrà un enginyer informàtic amb un perfil apte per a desenvolupar amb èxit els diversos aspectes tècnics del projecte.

Per últim, les figures d'un artista 3d i un animador serien necessàries, ja que caldrà un seguit d'animacions i un model 3d humà. Així i tot, aquestes figures són prescindibles; l'objectiu d'aquest projecte és desenvolupar uns algorismes en concret, els models i les animacions¹⁰ no més serveixen a efecte del testeig. Per aquest motiu i per estalviar recursos, es prescindirà de la contractació d'aquests dos treballadors.

La Taula 2 mostra els sous per hora bruts corresponents a cada empleat[21, 22, 23].

Posició	Sou brut hora	Sou brut anual
Recursos humans	10.55€	21,934€
Cap de projecte	68€	48,794€
Enginyer informàtic	15€	32,968€

Taula 2: Sou brut per hora treballada i anual.

La relació tasques-treballador-hores i cost final és expressada a la Taula 3. Com es pot apreciar, la majoria de tasques poden realitzar-se amb un sol empleat. Les excepcions, aquelles tasques relacionades amb el seguiment del projecte, on cal una coordinació entre el cap de projecte i l'enginyer. També cal destacar que s'ha creat una nova tasca per fer constar el procés de selecció i que s'han eliminat les tasques de documentació i recerca, ja que es realitzaran conjuntament amb la resta (un dels motius per les quals es van sobreestimar temporalment).

Remarcar que s'ha augmentat en un 30% el cost de les diferents activitats per considerar els costos associats amb la contractació dels empleats.

¹⁰A l'apartat de costos generals es donen més detalls del cost real dels models i animacions

Tasca	Cap de projecte	Recursos humans	Enginyer	Cost final (€)
Selecció personal	0	10	0	137.15
Abast	30	0	0	2652
Planificació	30	0	0	2652
Pressupost	30	0	0	2652
Sostenibilitat	8	0	0	312
Agile: Planificació general	20	0	20	156
Agile: Planificació prèvia	0	0	6	117
Agile: Test de qualitat	0	0	20	390
Agile: Reunió de control	6	0	6	647.4
Preparació entorn de treball	0	0	5	97.5
Sistema control <i>ragdolls</i>	0	0	50	975
Sistema cinemàtica inversa	0	0	25	487.5
Algorisme detecció pèrdua equilibri	0	0	30	585
Algorisme recuperació equilibri	0	0	100	1950
Algorisme detecció impactes imminents	0	0	30	585
Algorisme resposta autoconservació	0	0	20	390
Conclusió	23	0	0	2033.2
Defensa	23	0	0	2033.2
Total (CPA)	170	10	312	18851.95

Taula 3: Hores per tasca associada a cada treballador i cost final per tasca.

6.1.2 Costos genèrics

A continuació es descriuen tots aquells costos independents a les tasques del projecte:

- **Espai físic:** per a realitzar presencialment les diverses tasques del projecte. Es llogarà un espai en una oficina de *Coworking* per a una sola persona. El motiu pel qual no cal espai per a més persones és degut al fet que les tasques no es poden fer de forma simultània. El procés de selecció de personal serà anterior a l'arribada de qualsevol altre membre, per raons òbvies; els processos de gestió prèvia (Planificació, Abast, Pressupost, Sostenibilitat) s'han de realitzar abans les iteracions *Agile*; només coincidiran dos empleats durant les reunions de control que poden inclòs no ser presencials.
- **Ordinador:** pel mateix motiu, mencionat a l'anterior apartat, només en caldrà un. S'haurà de tenir en compte que hi haurà tasques exigents, computacionalment, que requeriran un *hardware* per sobre de la mitja. Com que els perifèrics tenen un cost molt baix en comparació, el seu cost serà afegit directament al de l'ordinador.
- **Software:** s'utilitzarà el mateix que s'ha mencionat en anteriors apartats. Això inclou: Unity3D, LaTeX, Python i Trello. Totes aquestes eines tenen un pla d'ús gratuït que s'adapta a les necessitats del projecte[24, 25, 26, 27].
- **Model 3D i animacions:** caldrà un model humà i unes animacions per a realitzar els experiments. Les animacions s'obtingran de forma gratuïta de Mixamo[28]. El model també serà obtingut gratuïtament de la *asset store* de Unity3D[29].

La Taula 4 mostra els costos genèrics identificats anteriorment[30], la seva amortització¹¹ i vida útil en mesos.

Element	Cost	Vida útil	Cost amortitzat
Espai <i>Coworking</i>	135€/mes	-	-
Ordinador	2100€	96	20.93€
Model 3d	0€	0	0€
Animacions	0€	0	0€
Unity3D	0€	0	0€
LaTeX	0€	0	0€
Python	0€	0	0€
Trello	0€	0	0€
Total (CG)	2640€	-	20.93€

Taula 4: Costos genèrics.

¹¹En el cas de l'espai de *Coworking*, no es té en compte el cost d'amortització, ja que és un actiu llogat. La resta d'actius, a excepció de l'ordinador, al tenir un cost inicial 0, la seva corresponent amortització acaba valorant-se en 0€

6.1.3 Costos associats al risc

Alguns riscos, en línia amb el mencionat a la planificació temporal, ja es troben assumits de forma implícita en el mateix còmput d'hores de les activitats. Igualment, existeixen tres setmanes com a marge de maniobra en cas que no es puguin complir els terminis, risc que definirem com a I1. Aquestes setmanes implicarien un altre més de lloguer de l'espai de *Coworking*, més hores de treball per part de l'enginyer i poques hores més per part del director (a fi de supervisar l'última iteració). Es considerarà un risc del 40%.

En resum, es considerarà un total de 60 hores repartides entre les tres setmanes addicionals. D'aquestes 60 hores, s'estima que un 100% les realitzarà l'enginyer i un 20% el cap de projecte.

6.1.4 Cost de contingència

S'ha decidit escollir un cost de contingència del 20% respecte al total de costos genèrics i activitats. Un motiu és, no tenir una clara solució a alguns dels problemes, el qual pot portar a l'afectació dels terminis. És cert que hi ha una sobreestimació dels temps amb les tasques més incertes i crítiques. Així i tot, es prefereix tenir un suport per cobrir el cost de risc anteriorment mencionat. L'altre motiu és la simplificació del cost que es farà del personal, el qual pot portar a una possible subestimació del cost total real.

6.1.5 Pressupost final

Reunint la informació de les Taules 2, 3 i 4 s'ha arribat al pressupost mostrat a la Taula 5. Recalcar que s'ha augmentat en un 30% el cost de les diferents activitats per considerar els costos associats amb la contractació dels empleats.

Concepte	Import (€)
Total CPA	18851.95
Total CG	2640
Total Costos (CG + CPA)	21491.95
Contingència	4298.39
CG + CPA + Contingència	25790.34
I1 (Cost=709.74; risc=40%)	283.896
Total imprevistos	283.896
Total	26074.236

Taula 5: Pressupost final.

6.2 Control de gestió

La presència de diferents obstacles, definits anteriorment, amenacen la fiabilitat d'aquest pressupost.

Un dels principals problemes que s'han mencionat és el fet que una o més de les iteracions fracassin. El cost de contingència i la partida de risc estan pensats per cobrir aquest cost en la mesura del possible. Si el període de tres setmanes no fos suficient, caldria llogar un mes addicional l'espai de *Coworking* i remunerar les hores extres als empleats; d'una manera semblant a l'expressada en el cost de risc I1.

Respecte als costos de hardware, trobar una alternativa gratuïta és impossible en cas de fallada. En aquest cas, s'intentaria reparar el hardware actual. Només en cas que fos fallada total es reemplaçaria per un nou hardware,

En el cas del software, el model i les animacions, és difícil que hi hagi un canvi que pugui alterar el cost. Hipotèticament podria passar que una o més de les llicències canviessin, inutilitzant les actuals i forçant a pagar una determinada quantitat. En aquest escenari hipotètic (i molt improbable) s'optaria per trobar alguna alternativa gratuïta. En el cas de Python i LaTeX és impossible que això succeeixi (les organitzacions responsables són sense ànim de lucre). En el cas de Unity3D, es podria escollir un altre motor gràfic com Unreal Engine o Godot. Respecte a les animacions i el model, a internet abunden moltes altres alternatives gratuïtes amb llicències d'ús molt obertes.

En conclusió, sembla que el pressupost és prou robust, ja que ha estat dissenyat al voltant del risc més gran que es pot trobar: l'incompliment dels terminis. Altres riscos analitzats, com una fallada de hardware, no suposen un increment considerable del pressupost en el pitjor dels casos. Tot i això, a fi de realitzar un seguiment més concret de les possibles desviacions, s'utilitzarà un indicador de desviació en el preu de la mà d'obra i un altre per a la diferència de consum on: d_p i d_e fan referència a aquestes desviacions respectivament, c_r i c_e als costos reals i estimats i h_r i h_e a les hores reals i estimades.

$$d_p = (c_r - c_e) * h_r$$

$$d_c = (h_e - h_r) * c_e$$

Aquestes estimacions es realitzaran de manera rigorosa entre iteracions *Agile* a fi de poder readaptar el desenvolupament de les iteracions posteriors. En aquest projecte, els conceptes de matèria primera i vendes no tenen cap rellevància, per tant no es contemplarà cap mena de desviació en aquests àmbits.

7 Sostenibilitat

7.1 Autoavaluació

Després de realitzar l'enquesta de sostenibilitat he vist que els meus coneixements relacionats amb la matèria són molt escassos. Algunes preguntes m'han fet adonar que mai tinc en compte els efectes col·laterals que una solució TIC pot tenir, tot i ser en molts casos dràstics.

Una altra cosa a destacar és el fet que, durant la confecció d'aquesta pràctica, en cap moment la sostenibilitat ha tingut cap mena d'importància. El criteri imperatiu en tot moment ha sigut obtenir el menor cost econòmic possible sense tenir en compte quin impacte podien tenir les diferents decisions preses en les diferents dimensions sostenibles.

En resum, considero que en molts casos només es veu el benefici a curt termini però es negligeixen les conseqüències futures. Cada cop és més evident l'impacte d'un estil de vida insostenible, de forma generalitzada, respecte al planeta, l'economia i la societat.

7.2 Dimensió econòmica

Respecte a la dimensió econòmica de la matriu de sostenibilitat, opino que he estimat el cost intentant ser el més realista possible. No obstant això, és cert que potser he sobreestimat excessivament alguns costos per preveure els possibles riscos.

Actualment els costos derivats de la solució més utilitzada al mercat són molt excessius. Sobretot pel que fa al personal; per culpa de el difícil que és integrar-lo. Una solució oberta, com la que es proposa en aquest projecte, permetria un estalvi considerable de personal i la viabilitat de nous projectes.

7.3 Dimensió ambiental

En cap moment s'ha tingut en compte l'impacte ambiental d'aquest projecte durant la presa de decisions. Ara bé, això no implica que aquestes decisions siguin perjudicials. Tot el contrari.

En primer lloc, al ser conscient que la gran majoria de tasques no es podien realitzar simultàniament, he aconseguit elaborar un pla de desenvolupament en el qual només cal un ordinador per a tot el projecte. Aquest petit canvi ja pot generar una millor sostenibilitat.

Un altre aspecte a destacar és el fet d'utilitzar un espai de Coworking. Utilitzant aquest espai, redueixo la despesa elèctrica, ja que aquesta no estaria destinada únicament al nostre projecte sinó que seria compartida.

7.4 Dimensió social

Considero que la realització d'aquest projecte m'aportarà una major determinació i rigor com a persona. M'ajudarà a ser més metòdic amb futurs projectes i a analitzar gran part de les dimensions presents en un problema. En definitiva, no crec que l'únic impacte sigui estrictament acadèmic o laboral, opino que moltes de les coses apreses em podran ser útils en altres aspectes de la vida.

No crec que la solució al meu problema millori considerablement la vida de ningú. La meua solució segurament sigui útil per la indústria de l'entreteniment però crec que hi ha sectors

que poden ser més prioritaris en la felicitat de les persones. Per tant, no considero que aquest projecte suposi una necessitat real en l'àmbit social.

8 Identificació de lleis i regulacions

Order Completed

Thank you for your order.

This Agreement between Mr. Albert Ortiz ("You") and Elsevier ("Elsevier") consists of your license details and the terms and conditions provided by Elsevier and Copyright Clearance Center.

Your confirmation email will contain your order number for future reference.

License Number 4834731145184 [Printable Details](#)

License date May 23, 2020

Licensed Content		Order Details	
Licensed Content Publisher	Elsevier	Type of Use	reuse in a thesis/dissertation
Licensed Content Publication	Graphical Models	Portion	figures/tables/illustrations
Licensed Content Title	FABRIK: A fast, iterative solver for the Inverse Kinematics problem	Number of figures/tables/illustrations	3
Licensed Content Author	Andreas AristidouJoan Lasenby	Format	electronic
Licensed Content Date	Sep 1, 2011	Are you the author of this Elsevier article?	No
Licensed Content Volume	73	Will you be translating?	Yes, without English rights
Licensed Content Issue	5	Number of languages	1
Licensed Content Pages	18		
Journal Type	S&T		

About Your Work		Additional Data	
Title	Síntesi d'animacions amb físiques i recuperació de l'equilibri	Portions	Fig. 1, Fig. 4, Fig. 7
Institution name	Polytechnic University of Catalonia	Specific Languages	Catalan
Expected presentation date	Jul 2020		

Requestor Location		Tax Details	
	Mr. Albert Ortiz	Publisher Tax ID	GB 494 6272 12

Requestor Location

Spain
Attn: Mr. Albert Ortiz

Figura 10: Rebut de la sol·licitud del permís de reutilització de les figures d'un article científic.

Gràcies a les característiques d'aquest projecte, les úniques lleis que hi poden tenir rellevància són les relacionades amb els drets d'autor.

El codi desenvolupat en el marc d'aquest TFG no utilitza cap llibreria externa. Ara bé, aquest codi va integrat dins el motor *Unity3D* i el seu ús si es troba regulat. Els termes d'ús permeten utilitzar *Unity3D* de forma totalment gratuïta per a finalitats acadèmiques[31].

També s'ha fet servir els models de *Microsoft Rocketbox*, una llibreria amb més de 115 avatars virtuals. Aquest any s'ha obert de forma gratuïta el seu accés al món acadèmic[32].

D'altra banda, respecte a la memòria, s'ha utilitzat algunes figures d'altres articles científics. Mitjançant el servei ofert per *Copyright Clearance Center*[33] s'ha pogut obtenir permís per a reutilitzar-les. A la Figura 10 es pot apreciar un dels rebuts.

9 Control de *ragdolls*

9.1 Definició del problema

Com ja s'ha comentat anteriorment, els *ragdolls* no tenen moviment propi; són simulacions físiques. El problema consisteix a trobar un algorisme capaç d'intervenir en aquestes simulacions per aconseguir que el cos realitzi un determinat moviment. Aquesta intervenció en cap cas pot anar en contra de les restriccions físiques imposades de base. Per exemple, si adoptar la postura resultant implica traspasar amb un braç una paret, aquest braç ha de respectar la col·lisió i complir parcialment la postura. Per tant, l'algorisme ha de ser capaç que el *ragdoll* pugui convergir en una postura determinada sempre que es respectin prioritàriament les limitacions físiques de l'entorn.

A la Figura 11 es pot apreciar l'escenari del qual es parteix com a base. A l'esquerra tenim el personatge sense cap mena de simulació física i amb una postura determinada. A la dreta tenim el seu corresponent *ragdoll*.

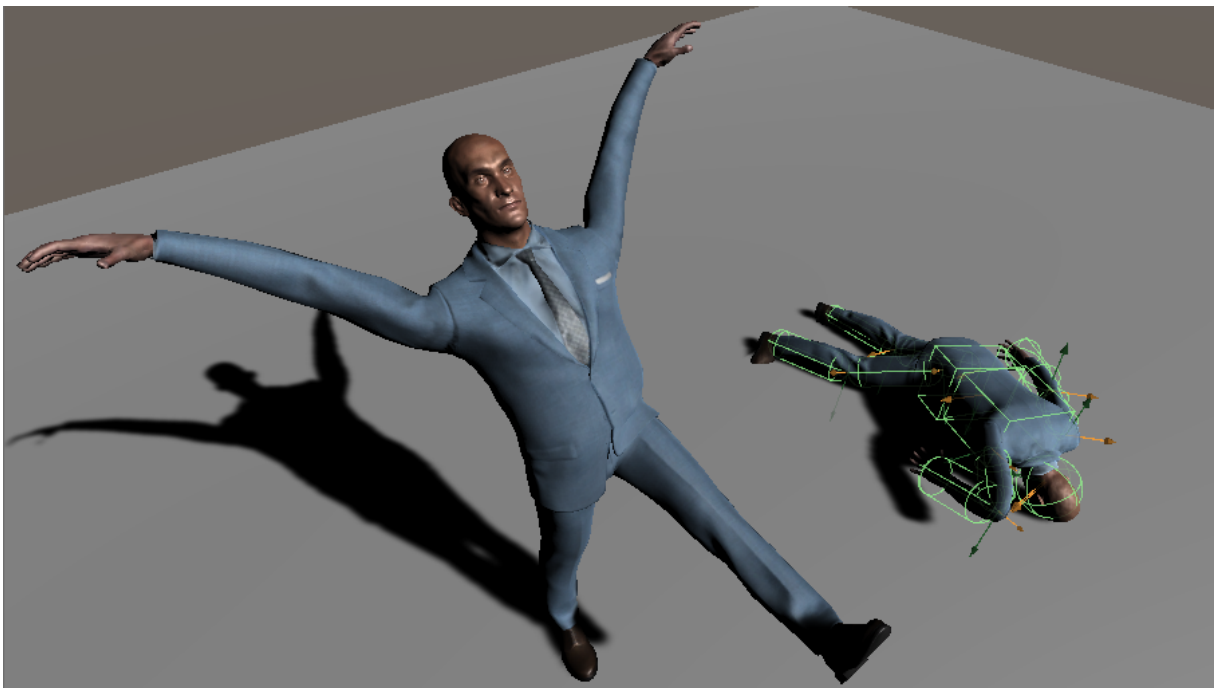


Figura 11: Escenari inicial

9.2 Solució

Els motors de físiques permeten intervenir en les simulacions, de forma directa, mitjançant forces de tota classe. En concret es poden aplicar dos tipus de forces: lineals i de torsió o torque. Ambdues es representen mitjançant un vector, no obstant això, tenen comportaments diferents. Les forces lineals són aquelles que empenyen l'objecte de forma directa en la direcció i magnitud indicades pel vector. Per tant, modifiquen únicament la velocitat lineal de l'objecte¹². Les forces

¹²Assumint que la força s'aplica directament sobre el centre de massa

de torsió en canvi, com el seu nom implica, apliquen una força que indueix una rotació sobre l'objecte. La direcció del vector, en aquest cas, indica l'eix sobre el qual es realitza la rotació i, la magnitud, tant el sentit com la intensitat. En la resolució d'aquest problema, aquestes forces són les que han tingut major rellevància.

Els quaternions també han tingut una gran importància a l'hora de resoldre aquest problema. Definim els conjunts $O = \{q_0, q_1, \dots, q_{n-1}\}$ i $R = \{q'_0, q'_1, \dots, q'_{n-1}\}$ com els conjunts que contenen els n quaternions que representen les orientacions objectiu i actuals de les n articulacions respectivament del cos del personatge.

En un principi es va considerar, per a tots els quaternions de R , interpol·lar-los esfèricament fins a assolir les respectives orientacions quaternions de R . La interpolació esfèrica es pot implementar mitjançant l'expressió[34]:

$$SLEP(q', q, t) = (q * q'^{-1})^t * q_0$$

El problema però, és que intervenir directament en una simulació física no sol produir gaires bons resultats. Això és degut al fet que, com que les simulacions físiques es produeixen en fragments de temps discrets, si entre fragment i fragment hi ha una alteració en la posició/orientació dels objectes a simular, el motor de físiques pot ser que no sigui capaç de detectar una possible intersecció entre cossos[35]. Això pot comportar que les restriccions físiques dels cossos es violin fàcilment i, per tant, les simulacions resultants acabin essent inestables. Una manera de solucionar això és establint els cossos associats al *ragdoll* com a *kinematics*. Dins de la terminologia de físiques per a videojocs, un cos *kinematic* és un cos físic al qual no l'afecten les simulacions físiques però, així i tot, la resta d'objectes poden seguir col·lidint amb ell. Certament això produiria que el *ragdoll* pogués seguir les animacions sense cap classe de problema. L'inconvenient, però, és que ja no estaria sotmès a les simulacions físiques. Per tant, tornariem a estar al punt de partida.

Una manera de solucionar això és amb el següent procediment: si el motor de físiques detecta un impacte associat a un dels cossos del *ragdoll*, aquests cossos i els seus fills deixaran de ser *kinematics* i estaran totalment sotmesos a les simulacions físiques. Tampoc seran interpolats. Després d'un temps, tornaran a ser *kinematics* i interpolats novament. L'inconvenient de tot aquest procés, com es pot apreciar, és que és bastant enrevessat i els resultats disten molt de ser acceptables; el fet que en alguns casos una part del cos es trobi seguint l'animació sense tenir en compte les forces externes i unes altres es trobin totalment sotmeses a simulació, fa que les parts sotmeses a la simulació acabin tenint velocitats molt caòtiques. A més a més, i lligat a aquest últim problema, si per exemple una de les articulacions es troba sota simulació física, no hi ha manera que l'extremitat pugui seguir l'animació parcialment utilitzant interpolacions tot i estar sotmesa a simulacions.

El mètode que es proposa en aquest TFG supera tots els inconvenients de les alternatives esmentades anteriorment. Aquest mètode aconseguix que un *ragdoll* sigui capaç d'imitar el moviment d'una animació utilitzant únicament el resultat de simulacions físiques.

La solució proposada consisteix, en primer lloc, a obtenir les orientacions relatives dels quaternions de R respecte O . Podem obtenir fàcilment l'orientació relativa entre dos quaternions $q \in O$ i $q' \in R$ mitjançant l'expressió:

$$\Delta Q = qq'^{-1}$$

ΔQ pot ser vist com el quaternió que hem d'aplicar a q' per assolir l'orientació de q . Seguida-

ment, un cop obtingut ΔQ , el podem descompondre en un eix i un angle utilitzant una extensió de la fórmula d'Euler[36]:

$$\Delta Q = \cos\theta/2 + (u_x\mathbf{i} + u_y\mathbf{j} + u_z\mathbf{k})\sin\theta/2$$

De forma similar a [37], s'aplica una força de torsió a cada un dels cossos que formen el *ragdoll*, utilitzant l'eix i l'angle extrets del seu ΔQ corresponent i un factor k per a regular la intensitat de la força final. Amb un valor $k = 3.91$ s'ha assolit resultats prou satisfactoris.

Un inconvenient que s'ha trobat a l'hora de desenvolupar aquesta solució és el fet que les articulacions vibraven abans d'assolir l'orientació objectiu. Experimentant es va determinar que aquest comportament era produït per l'absència de fricció, tant angular com lineal. Això és degut al fet que, sense fricció, la força aplicada pot superar fàcilment l'orientació desitjada. Un cop superada aquesta orientació objectiu, l'algorisme tornarà a aplicar una nova força en sentit contrari que farà que, novament, es torni a superar l'orientació objectiu; així successivament fins a convergir en l'orientació final. Augmentant la fricció, s'aconsegueix filtrar aquest moviment oscil·latori. En concret, amb valors de $f_L 11.6$ i $f_W = 36.6$ de fricció lineal i angular respectivament, s'ha assolit bons resultats.

A la Figura 12 es pot apreciar el personatge sense físiques amb la postura objectiu (esquerra) i el *ragdoll* intentant imitar-la (dreta). No només la postura resultant del *ragdoll* és més natural, sinó que les parts del seu cos s'adapten a la geometria de l'entorn per evitar travessar-la. En aquest cas, la cama dreta del *ragdoll* s'ha flexionat automàticament per evitar traspasar el terra.

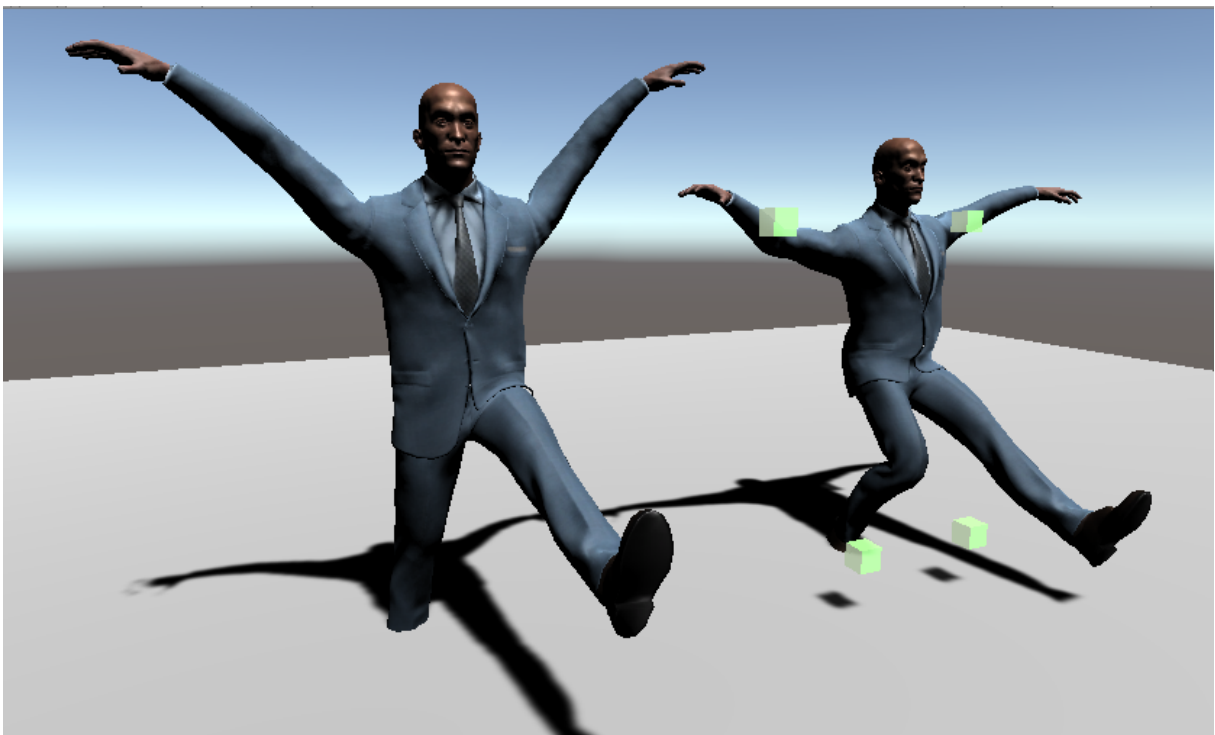


Figura 12: Resultat final

9.3 Treball futur

Tot i que el resultat final ha sigut prou satisfactori, han quedat coses per millorar. En primer lloc, per animacions lentes el resultat es veu prou bé. En canvi, per animacions ràpides, el moviment resultat és bastant artificial. Caldria implementar algun sistema que regulés la força de torsió en funció de la velocitat angular de les articulacions de l'animació objectiu.

Per últim, el *ragdoll* no assoleix l'estabilitat de forma natural. Com ja s'ha comentat, les simulacions que es realitzen en un videojoc són molt simplificades. És per això que en molts casos on el *ragdoll* hauria d'estar en equilibri, cau. Una manera de solucionar això, tot i que físicament irreal, és desactivant la velocitat lineal del centre de massa del *ragdoll*, com també es fa a [37]. Controlar quan realment es perdrà l'equilibri i, per tant, s'activarà la velocitat lineal del centre de massa, és una tasca que es soluciona a la secció 11.

10 Cinemàtica inversa

10.1 Definició del problema

A l'apartat 1.1.3 s'ofereix una introducció bàsica al problema. En aquest TFG, però, s'ha hagut de tractar una instància del problema més difícil. Es busca poder trobar un algorisme de cinemàtica inversa que permeti treballar amb quatre *end-effectors* simultàniament (un per a cada extremitat). El sistema ha de ser capaç de trobar un angle per a cada articulació, respectant les òbvies restriccions biomecàniques, a fi que es minimitzi la distància total dels *end-effectors* amb els seus respectius objectius. El cos del personatge s'ha simplificat a 2 articulacions per extremitat (no es té en compte canells ni turmells) i 4 articulacions per l'esquena. Si tenim en compte només els angles de rotació perpendiculars a l'extremitat i que 4 de les articulacions només poden girar sobre un eix, això ens deixa amb un total de 20 graus de llibertat. Tot i les simplificacions, la dimensió del problema segueix essent problemàtica.

La posició dels *end-effectors* pot ser definida com a $s = f(\Theta)$ on s és una matriu que conté la posició dels *end-effectors* (en aquest cas una matriu 3×4) i Θ un vector columna que conté els 20 graus de llibertat. Per tant, idealment el que volem és invertir f per a obtenir Θ en funció de s . Cal tenir en compte que, a l'estar f definida a partir de matrius de translació i rotació, conté expressions trigonomètriques i, per tant, no és lineal. Això, la dimensió del problema i el fet que poden existir una, varies o cap solució són els principals factors de dificultat a l'hora de trobar $\Theta = f^{-1}(s)$.

És necessari posar especial èmfasi en les restriccions biomecàniques que el sistema ha de ser capaç de respectar. A la Figura 13 es mostren els tipus d'articulació més comuns en un esquelet humà. En el cas d'aquest TFG, només es tindrà en compte els *Hinge* i *Ball-and-Socket*; la resta d'articulacions afegirien una complexitat innecessària al model. Els *Hinge* són de les articulacions més restrictives. Com el seu nom indica, el moviment que poden permetre és l'equivalent al d'una frontissa. Exemples en el nostre cos: colzes i genolls. En canvi, el *Ball-and-Socket* és un dels tipus d'articulació que permet major rang de moviment; permet un moviment rotatori entre dos eixos simultanis. En aquest cas, els malucs i les espatlles són clars exemples. A part de les restriccions axials, cap articulació està lliure de restriccions angulars[38].

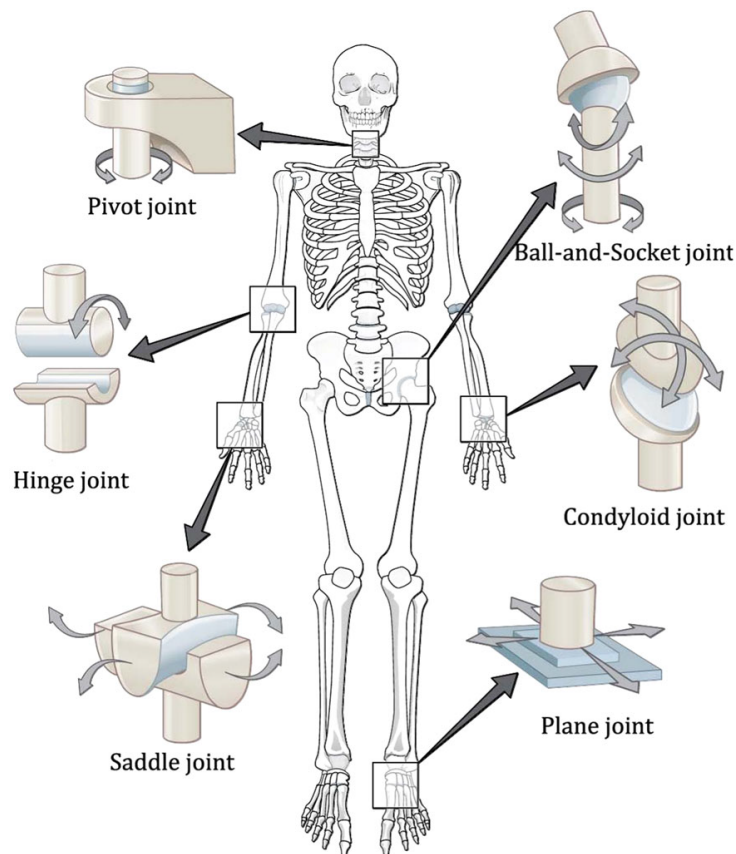


Figura 13: Els sis tipus d'articulació més comuns al cos humà

Font: <http://anatomisty.com/>

10.2 Solució

A causa de la naturalesa d'aquest problema i a les diferents configuracions i casos d'ús possibles, existeixen diverses famílies de mètodes resolutius. A fi de justificar la solució emprada, primerament s'ofereix una breu descripció de les metodologies més típiques.

10.2.1 Mètodes analítics

La idea fonamental d'aquesta família consisteix a explotar la geometria del sistema a tractar, derivant equacions per a trobar Θ ; principalment utilitzant trigonometria. Amb aquests mètodes generalment es pot trobar totes les solucions possibles, són fiables i computacionalment barats[39]. El principal inconvenient és que, a mesura que els graus de llibertat del problema augmenten, derivar una solució analítica resulta molt més difícil i inclòs pot arribar a ser impossible[40]. Actualment, s'ha aconseguit trobar una solució analítica per al *6R manipulator*, un sistema articulat amb 6 graus de llibertat[41]. A la Figura 14 es pot apreciar aquest sistema.

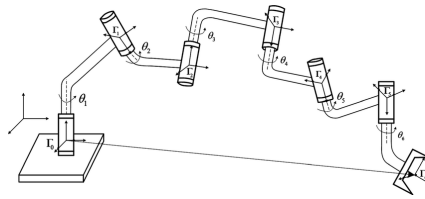
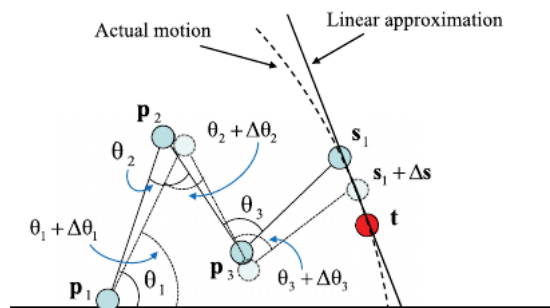


Figura 14: Esquema del 6R manipulator

Font: Zhongtao Fu

10.2.2 Mètodes numèrics

Aquests mètodes són tots aquells que requereixen d'un nombre d'iteracions per confluïr a una solució. Aquesta família es pot subdividir en tres categories: *Jacobian*, *Newton* i *Heuristic*[42]. Els mètodes *Jacobian* utilitzen les derivades parcials de la cadena d'articulacions respecte Θ . Aquestes derivades parcials formen la matriu *Jacobian* J . La idea bàsica és utilitzar aquestes derivades com aproximació lineal de la posició final de l'*end-effector* i, mitjançant petits $\Delta\Theta$, intentar minimitzar la distància entre l'*end-effector* i l'objectiu. A la Figura 15 es mostra un exemple de l'aproximació.

Figura 15: Aproximació lineal utilitzant la *Jacobian matrix*

Font: Computer Graphics Forum

Utilitzar un mètode *Jacobian* implica computar J^{-1} , lo qual en molts casos no és possible, ja que J no sempre és quadrada. Per aquest motiu, sorgeixen moltes subcategories (pseudoinversa, transposada, *singular value decomposition*, ...) que afronten de diferents maneres el problema d'aproximar J^{-1} . Entrar en detall amb aquests mètodes s'escapa dels objectius del TFG. Cal destacar, però, que molts d'aquests mètodes solen patir el problema de la singularitat; cas on cap $\Delta\Theta$ té efecte sobre l'*end-effector*. Un altre gran problema és que solen trigar molt a trobar una solució i el moviment articulari resultant no sol ser gaire natural[43].

Els mètodes de *Newton* aproximen la funció objectiu utilitzant el polinomi de Taylor de segon ordre i optimitzant-la amb el mètode *Quasi-Newton*. Un dels principals punts a favor d'aquests mètodes és que, al tractar-se d'una constant optimització, el moviment articulari resultant és suau. Un altre aspecte a destacar és la facilitat d'incorporar noves restriccions al sistema, no només angularment, com es mostra a [18]. També cal destacar que aquests mètodes no pateixen el problema de la singularitat. Malgrat això, cal calcular la matriu Hessiana i la seva inversa, el qual és computacionalment molt costós. Aquest cost impacta greument al temps de cada

iteració. A part, solen ser sistemes complexos i difícils d'implementar.

Finalment, tenim els algorismes Heurístics. Es caracteritzen per resoldre el problema de cinemàtica inversa arribant a la solució de forma iterativa sense utilitzar equacions o càlculs complexos. Generalment ofereixen un cost computacional molt menor respecte la resta de mètodes numèrics. Un dels principals desavantatges és que el moviment resultant, en molts casos, pot ser antinatural ja que cada articulació es tracta de forma individual. D'aquesta família d'algorismes tenen especial rellevància *Forward and backward reaching inverse kinematics (FABRIK)* [44] i *Cyclic coordinate descent (CCD)* [45]. És important destacar que *FABRIK* permet resoldre cossos multiarticulats mentre que *CCD* només permet resoldre una sola cadena.

10.2.3 Mètodes basats en dades

Aquests mètodes afronten el problema de la cinemàtica inversa utilitzant grans bases de dades de captura de moviment. La idea fonamental és, d'entre un conjunt de postures apreses, trobar la que té els *end-effectors* més propers a les posicions objectiu. Generalment se solen utilitzar algorismes basats en xarxes neuronals. El principal avantatge d'aquests algorismes és que sempre es generen postures molt fidels a la realitat. Un desavantatge, que es va mitigar amb PCA¹³, és el cost computacional per trobar la postura més similar entre un conjunt de candidats tan gran [46]. Tot i això, en aquests mètodes el principal inconvenient és la dificultat que la seqüència de postures generada tingui coherència; diferents postures poden complir les restriccions dels efectors i així i tot no tenir cap sentit respecte a la postura anterior. El resultat final sempre estarà totalment condicionat amb la qualitat i quantitat d'aquestes dades [42].

10.2.4 Mètode emprat

El temps de còmput i la resolució de postures per a cossos multiarticulats han sigut els criteris decisius a l'hora d'escollir un mètode. Per tant, un mètode heurístic com *FABRIK*, semblava ser el candidat més ideal.

La idea bàsica de *FABRIK* és simple: en comptes de reorientar les articulacions de forma iterativa, es desplacen directament en dues fases (anomenades *Forward* i *Backward reach*, d'aquí el nom de l'algorisme) i, posteriorment, es reorienten. A la Figura 16 es pot veure esquemàticament el funcionament d'una iteració completa, amb una cadena de 4 articulacions i un únic objectiu:

- (a) Mostra la configuració inicial amb el punt t com l'objectiu al qual ha d'arribar p_4 .
- (b) p_4 es desplaça directament a la posició objectiu, esdevenint p_4'
- (c) Es defineix una línia entre p_3 i p_4' sobre la que es projecta p_3 (esdevenint p_3' tenint en compte la longitud de l'extremitat que connecta $p-3$ i $p-4$, d_3).
- (d) Mostra l'anterior pas propagat a la resta d'articulacions fins a arribar a l'arrel.
- (e) En aquest pas es mostra el començament de la segona fase de *FABRIK*. Aquesta segona fase consisteix a repetir els passos (a-b-c), en sentit invers i establint com a objectiu de

¹³Principal component analysis

l'arrel la seva posició inicial: ja que aquesta segurament s'haurà mogut per les accions de la primera fase.

- (f) Mostra el resultat de l'última fase. Com es pot veure, l'*end-effector* es troba a la posició objectiu i l'arrel a la seva posició original. L'algorisme repeteix aquestes dues fases de forma iterativa fins que, o bé s'ha assolit un màxim d'iteracions o bé s'ha superat una distància mínima entre l'*end-effector* i l'objectiu.

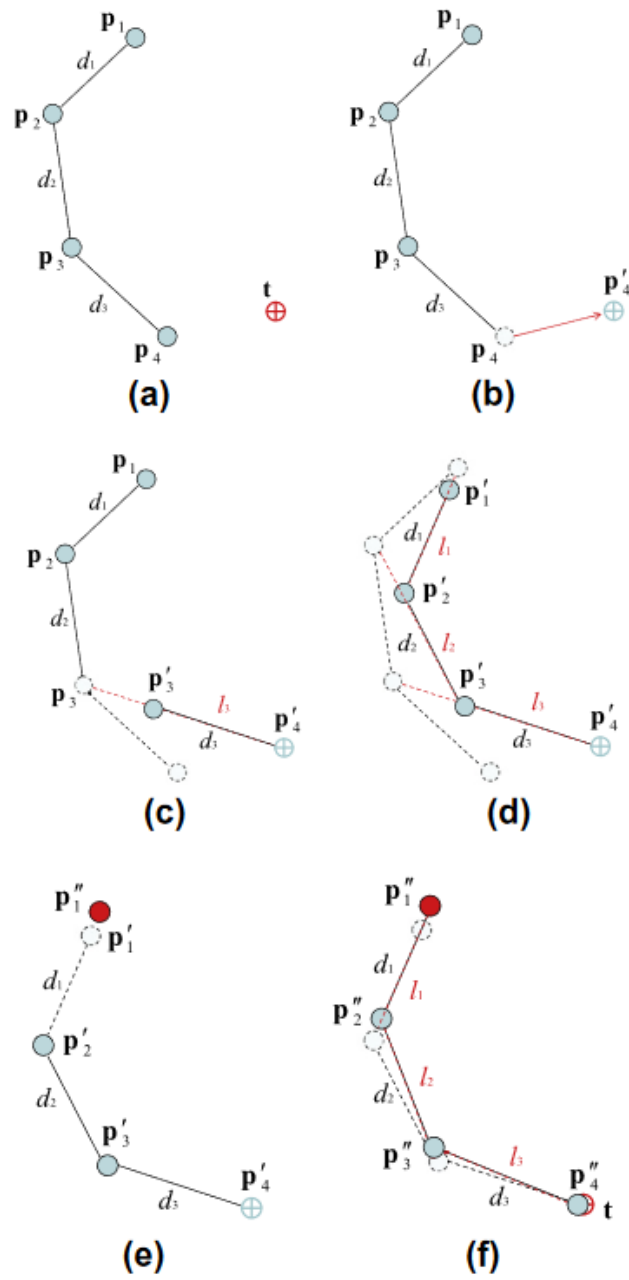


Figura 16: Iteració completa de FABRIK

Font: Elsevier Inc.

Implementar FABRIK per una sola cadena és molt senzill. Estendre-ho a un cos sencer, sorprenentment, tampoc és complicat. De fet, també es pot incloure sense gaire complicació objectius per articulacions que no siguin fulles¹⁴. On realment es complica el procés d'implementació és a l'hora d'afegir restriccions. Aquesta tasca és la que més ha perjudicat el desenvolupament del projecte. El principal problema és que hi ha diversos tipus de restriccions biomecàniques a respectar, com ja s'havia previst a la definició del problema. L'article original on s'introdueix FABRIK [44], exposa que l'algorisme es capaç d'incorporar restriccions de tot tipus i en dona una orientació bàsica per les restriccions més típiques. La idea per afegir restriccions consisteix a reubicar p_{i-1} , envers p_i , segons les restriccions que es vulgui implementar, un cop s'ha realitzat el pas (c) de l'algorisme.

El tipus de restricció influeix dràsticament en la reubicació de p_{i-1} , com es pot veure a la Figura 17. Per tal d'implementar la restricció a dalt (*Ball-and-Socket*), cal primer projectar p_{i-1}' sobre la línia L_1 formada per p_i i p_{i+1} . D'aquesta projecció obtenim S (en relació a p_i). Com que un *Ball-and-socket joint* no sol tenir les mateixes restriccions angulars en ambdós eixos, de fet ni en els dos sentits de rotació sobre un mateix eix, tenim: $\Theta_1, \Theta_2, \Theta_3$ i Θ_4 com a angles de rotació. Aquests angles descriuen la regió cònica que es pot apreciar a la Figura 17, part superior (a). Podem obtenir la longitud dels eixos d'aquesta regió q_1, q_2, q_3, q_4 mitjançant l'expressió $q_j = S \tan \Theta_j$. Amb la longitud d'aquests eixos, per a cada quadrant es pot definir una el·lipse diferent que descriurà la projecció sobre el pla amb normal L_1 de la regió cònica irregular. Un cop obtinguda aquesta projecció, cal també projectar p_{i-1} en aquest mateix pla, trobar a quin quadrant pertany i posteriorment calcular el punt més proper a l'el·lipse corresponent (el qual no és trivial). Aquest punt proper, un cop a l'espai original, serà la nova posició de l'articulació dins dels límits establerts per les restriccions de l'articulació anterior.

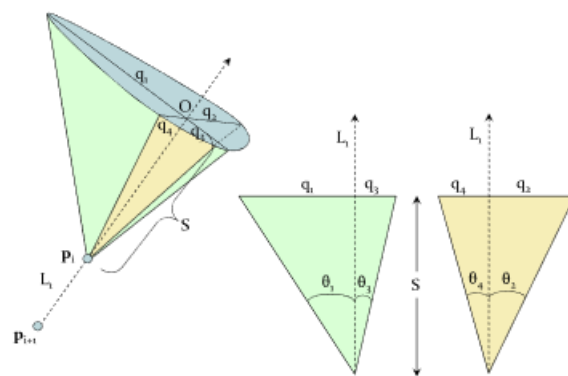
Per implementar una restricció de tipus *Hinge*, cal tenir en compte que, si p_i és l'articulació d'aquest tipus, p_i ha de romandre sempre en el pla format per p_{i-1} i p_{i+1} . A la Figura 17 es pot apreciar com a la part inferior (d) p_2' ha esdevingut p_2'' al projectar-se al pla Φ_2 .

Un cop es va implementar aquestes dues restriccions, el funcionament de l'algorisme va empitjorar dràsticament. Les solucions respectaven les restriccions imposades però no convergien en solucions òptimes¹⁵. A més a més, el moviment resultant era totalment erroni i discontinu. És a dir, si l'algorisme convergia en una determinada solució i l'objectiu es desplaçava un petit Δ , la nova solució en molts casos produïa un nou posicionament articulari totalment diferent a l'anterior. En un principi es va pensar que era un error d'implementació de les mateixes restriccions. Per descartar aquesta hipòtesi, es van executar els algorismes de restriccions en entorns de prova aïllats. Un cop descartada aquesta possibilitat, es va plantejar el fet que el nombre d'iteracions no fos suficient¹⁶.

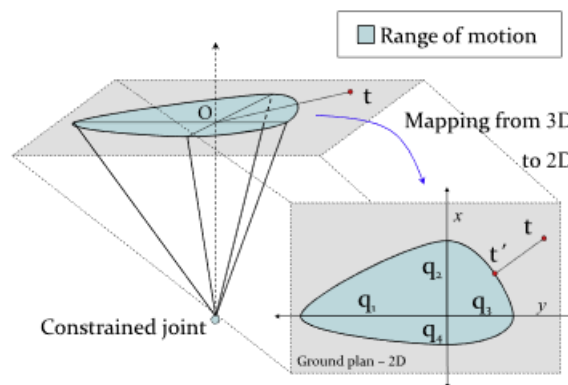
¹⁴Els esquelets dels personatges tenen una relació d'ordre en forma d'arbre; l'os de la pelvis és l'arrel i, els peus, el cap i les mans, les fulles. Les transformacions geomètriques d'un pare es propaguen als seus fills

¹⁵En situacions on l'objectiu era perfectament assolible, l'algorisme posicionava l'*end-effector* bastant lluny

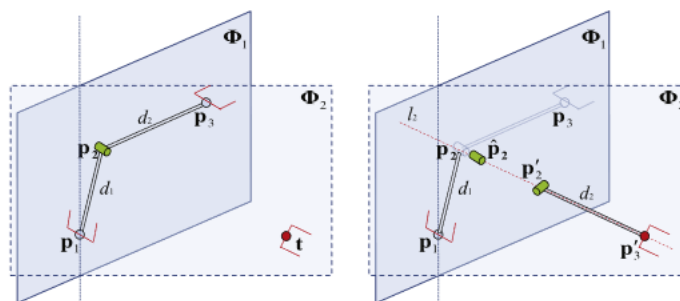
¹⁶Sense restriccions, FABRIK era capaç de convergir en una solució en menys de 10 iteracions, en la majoria de casos



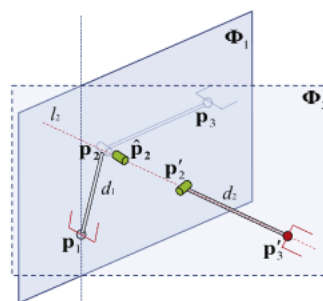
(a)



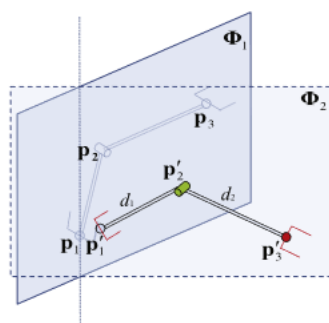
(b)



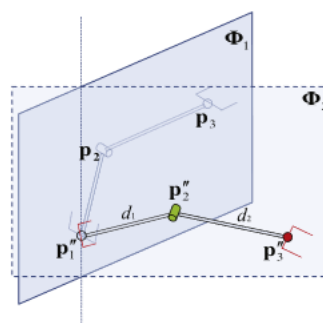
(a)



(b)



(c)



(d)

Figura 17: Com re-posicionar una articulació per complir restriccions de tipus *Ball-and-socket* (adalt) i *Hinge* (abaix)

Font: Elsevier Inc.

Augmentar el nombre d'iteracions no va oferir millors resultats, només va empitjorar dràsticament el temps de còmput. Després d'una extensiva cerca d'errors d'implementació, es va deduir que possiblement el problema tenia a veure amb l'algorisme en si. Es va concloure, per tant, que l'eficàcia de *FABRIK* en cadenes fortament restringides es veu greument afectat.

Un altre problema que s'ha hagut d'afrontar durant el desenvolupament d'aquest algorisme va ser el fet que, dins la jerarquia d'ossos d'un personatge, cada cadena interior té un sistema de referència totalment diferent¹⁷. La solució ha consistit a definir manualment els eixos d'orientació per a cada part del cos, a fi d'estandarditzar els sistemes de referència.

Finalment, s'ha decidit utilitzar una combinació de dos mètodes. En un principi, els mètodes analítics s'havien descartat perquè és impossible derivar-ne un en un sistema amb tants graus de llibertat. Ara bé, en cadenes curtes com braços i cames on *FABRIK* té problemes de convergència, els mètodes analítics resulten idonis. En primer lloc, com ja s'havia comentat, al tenir pocs graus de llibertat derivar una solució és fàcil. En segon lloc, aquests mètodes són els més eficients a nivell de recursos i temps de còmput total¹⁸. Finalment, són consistents i el moviment resultat és suau i natural. La idea bàsica del mètode analític utilitzat en aquest TFG consisteix a tractar una extremitat com un triangle, com es pot veure a la Figura 18. d_1 i d_2 són les respectives longituds de la part superior i inferior de l'extremitat, Θ_1 i Θ_2 són els angles que volem. L'algorisme bàsicament utilitza la llei de cosinus per a obtenir ambdós angles. Per a restringir el nombre solucions a una sola i el domini del problema a dues dimensions, s'utilitza una posició auxiliar com a referència. El producte creuat entre el vector que va de l'origen de l'extremitat superior a l'articulació i el vector que va des del mateix origen fins a la posició auxiliar defineixen la normal del pla sobre la que es resol el problema. El pla on es resolgui el problema, determinarà en última instància la posició del colze o el genoll. Per tant, aquesta posició es reubica automàticament per a mantenir sempre la mateixa posició relativa a l'articulació. D'aquesta manera, l'articulació mai viola les restriccions biomecàniques.

¹⁷Per exemple, des de l'espai global, l'eix x de l'espai del colze potser apunta cap a la següent articulació mentre que a l'espai del maluc potser és l'eix $-y$ el que apunta a la següent articulació

¹⁸En comparació amb els mètodes Heurístics, els analítics resolten el problema en una única iteració; sovint requerint càlculs trigonomètrics senzills

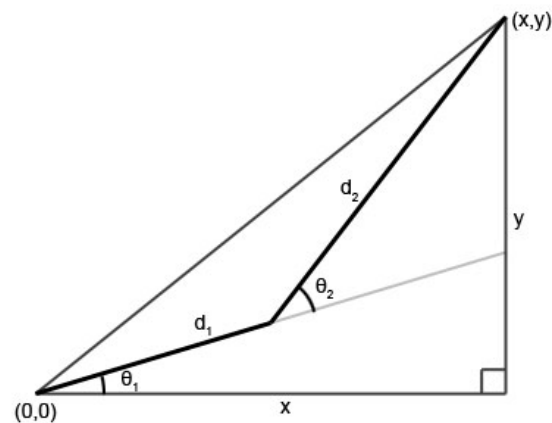


Figura 18: Transformació del problema de la cinemàtica inversa, per a una cadena de dos ossos, a un problema trigonomètric bidimensional.

Font: <http://ryanjuckett.com/>

Ara bé, amb això no es resol el problema; volem aplicar cinemàtica inversa al cos sencer. Aquí és on entra en joc *FABRIK*. Per a cadenes llargues i amb restriccions senzilles *FABRIK* resulta ser més viable que un mètode analític, per aquest motiu s'ha decidit utilitzar aquest algorisme per resoldre la columna vertebral¹⁹. L'algorisme tracta la cadena formada per l'ós arrel (la pelvis) fins al coll del personatge (que serà l'*end-effector*), utilitzant únicament restriccions de tipus *Ball-and-socket* entre vèrtebres. Quan es detecta un canvi en la posició dels objectius dels braços, l'algorisme recalcula l'objectiu d'aquesta cadena com el centroid de les dues posicions. A la Figura 19 es pot veure el resultat final. Les esferes verdes representen els objectius dels efectors. Les línies blanques mostren els triangles sobre els quals es deriva la solució analítica.

¹⁹Vegis apèndix A Codi 1, per a la implementació de *FABRIK* i Codi 2 per a la implementació de les restriccions

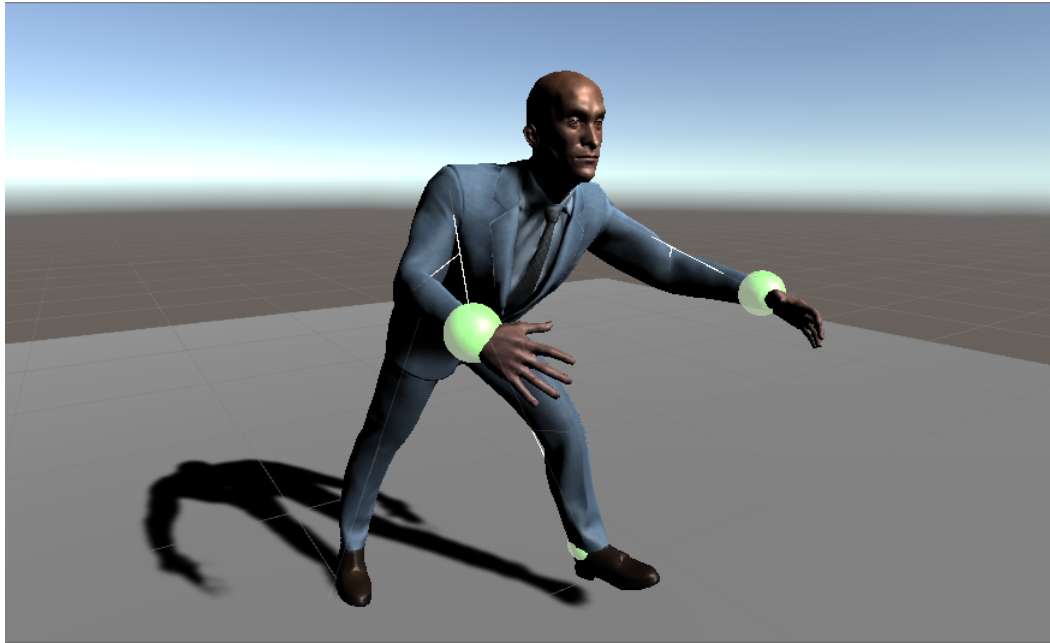


Figura 19: Resultat de l'algorisme de cinemàtica inversa. El personatge és capaç d'adoptar dinàmicament la postura que millor li permeti arribar als objectius (esferes verdes).

10.3 Treball futur

Tot i que des de la primera implementació de *FABRIK* el resultat ha millorat considerablement, encara queden coses per millorar. Utilitzar un mètode analític per les extremitats ha resultat ser una bona idea. S'ha pogut comprovar de primera mà els avantatges.

Ara bé, potser es podria haver utilitzat un altre mètode per resoldre la columna vertebral. En alguns casos el moviment produït per *FABRIK*, com ja s'havia comentat, produeix certs moviments bruscos i poc realistes. Una solució que no s'ha pogut verificar, consistiria a filtrar els pròxims increments de rotació indicats per l'algorisme, segons un increment límit extret de dades de moviment reals; a fi d'evitar increments bruscos i antinaturalment.

Queda pendent avaluar com *CCD* funcionaria en comptes de *FABRIK*. Un altre millora, el posicionament de l'objectiu de la columna. L'algorisme actual el recalcula constantment, es podria recalculer només quan no fos possible arribar a un dels objectius (calculant la longitud de l'extremitat i comparant-la amb la distància a l'objectiu). Això evitaria moviments innecessaris de la columna.

11 Pèrdua d'equilibri

11.1 Definició del problema

Com s'ha comentat anteriorment a la secció 9, les simulacions físiques d'una aplicació en temps real són molt aproximades. Per tant, encara que el *ragdoll* sigui capaç d'imitar postures on, en condicions reals, una persona mantindria l'equilibri, en molts casos caurà a causa de l'error present. També s'ha comentat que una manera de pal·liar aquest efecte és no simular el moviment rectilini del centre de massa. D'aquesta manera, el personatge es troba "flotant" però les forces encara afecten la resta d'extremitats i poden arribar a induir velocitat angular al centre de massa (d'una manera similar a com si estigués penjant d'un ganxo de carn).

El problema que deriva d'aquesta solució és determinar quan s'ha d'activar o desactivar la simulació de moviment rectilini del centre de massa per simular la pèrdua d'equilibri. En el següent apartat es proposa un mètode que ha donat bons resultats.

11.2 Solució

Calcular l'estat del centre de massa d'un cos humà, de forma físicament realista, es troba fora l'abast d'aquest TFG. L'objectiu és trobar un sistema poc costós en l'àmbit computacional i que generi resultats prou convincents. Per tant, s'ha descartat totes aquelles solucions que requereixin càlculs molt complexos. També s'ha descartat solucions excessivament simplificades com a [37], on només es té en compte l'orientació del centre de massa²⁰, però es negligeix l'estat de la resta d'articulacions (que també pot influir considerablement en la pèrdua d'equilibri).

La solució que es proposa en aquest TFG es basa en la idea d'utilitzar un polígon de suport, format pels punts de contacte del personatge amb el terra, i la massa assignada a cadascun dels cossos que formen el *ragdoll*. Perquè un cos es trobi en equilibri, una condició necessària és que el seu centre de massa es trobi dins el polígon de suport[47]. La idea de l'algorisme proposat consisteix a simplificar el cos del personatge en un conjunt de segments i computar un indicador que anomenarem FE²¹. Aquest valor, comprès en el rang $[0, 1]$ determinarà, en última instància, com de dins del polígon de suport es troba el centre de massa. Un valor de 1 significarà que el centre de massa es troba totalment dins el polígon de suport i un valor de 0 totalment fora. Quan el FE superi un cert llindar, s'activarà la simulació del moviment rectilini del centre de massa i el *ragdoll* caurà.

L'algorisme per a calcular el FE consisteix en:

1. Generar una figura convexa K a partir dels punts de suport del personatge (en aquest cas els peus, formant un paral·lelogram).
2. Per a cada cos i que forma el *ragdoll*, generar un segment s_i des del seu pivot fins al seu fill i assignar-li un pes m_i (la massa del cos). Això aproxima la geometria del cos del personatge.
3. Projectar K i s_i en el pla de la superfície de contacte, obtenint K' i s_i' respectivament. Així el problema es redueix a un problema bidimensional.

²⁰En coordenades globals

²¹Factor d'equilibri

4. $\forall i$, utilitzant test d'orientació en dos dimensions, classificar i segons:

(a) $s_i' \cap K' = \emptyset$

(b) $s_i' \cap K' = s_i'$

(c) $s_i' \cap K' \subset s_i'$

5. Sigui m_i la massa associada a un cos i i m_k la massa acumulada en K' (que inicialment val 0), $\forall i$ incrementem m_k segons els casos anteriors:

(a) $m_{k+} = 0$

(b) $m_{k+} = m_i$

(c) $m_{k+} = m_i(1 - |s_i''|/|s_i'|)$ on s_i'' és la part del segment s_i' que es troba fora de K' . Per a trobar s_i'' cal computar els punts d'intersecció entre $Ch^{22}(K)$ i K' tenint en compte que com a mínim hi pot haver un i com a màxim dos. En cas que sigui un únic punt, $|s_i''|$ és la distància euclidiana entre el punt d'intersecció i el punt que es trobi dins de K . Altrament, $|s_i''|$ serà la distància euclidiana entre els dos punts d'intersecció.

6. Per a tots els segments $s_i' \cap K' \subseteq s_i'$, sigui s_i'' la part de s_i' que es troba fora de K' i $b_{0,\dots,3}$ quatre comptadors inicialment a zero, s'afegirà la porció de massa de s_i'' , $m_i(|s_i''|/|s_i'|)$, a un dels comptadors segons l'expressió:

$$c = d \cdot \frac{|s_i'' - c|}{\|s_i'' - c\|}$$

on d és el vector unitari que indica la direcció a la qual es troba orientat el personatge i c el centroide del polígon de suport. S'escollirà el comptador b_i per a ser incrementat segons:

$$i = \begin{cases} 0, & \text{si } c \in [-1.0, -0.5) \\ 1, & \text{si } c \in [-0.5, 0.0) \\ 2, & \text{si } c \in [0.0, 0.5) \\ 3, & \text{altrament} \end{cases}$$

7. Seguidament, un cop incrementats els comptadors $b_{0,\dots,3}$, s'actualitzarà m_k amb la següent expressió:

$$m_{k+} = 2\min(b_0, b_3) + 2\min(b_1, b_2)$$

8. Un cop calculat m_k , podem computar FE mitjançant l'expressió:

$$FE = m_k/m_t$$

on m_t és la massa total acumulada de tots els cossos del *ragdoll*.

Els passos 1 – 5 computen la massa del personatge que incideix sobre el polígon de suport de forma directa, els passos 6 i 7 aproximen la massa que es compensa en caure en cantons oposats

²²convex hull

del polígon de suport. Concretament, el pas 6 distribueix els segments que es troben fora de K' en quatre quadrants, orientats en la direcció a la qual mira el personatge, utilitzant el producte escalar. El pas 7 afegeix a la massa que es troba dins de K' , m_k , la quantitat de massa que tenen en comú els quadrants oposats.

A la Figura 20 es pot apreciar el primer pas per a calcular el FE. En verd apareixen els segments s_i , en blau $Ch(K')$ i en vermell $s_i l$.

A la Figura 21 es pot veure la classificació dels segments. En verd, els segments que es troben dins de K , en vermell els que es troben fora i en groc els que es troben parcialment dins de K .



Figura 20: Projecció de K i s_i

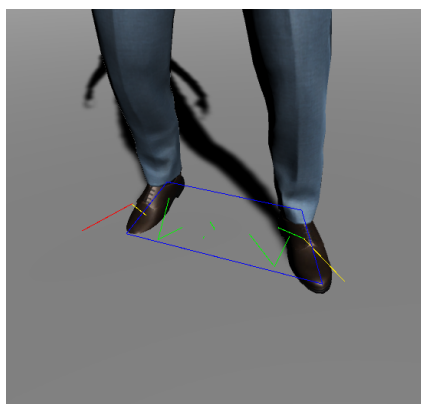


Figura 21: Classificació dels segments

Un cop calculat FE^{23} , només cal comprovar si el seu valor és inferior a un cert llindar c . Tot i que $c = 0.5$ seria el llindar més raonable, per a animacions extremes²⁴ és possible que es produeixi un fals positiu. Una possible solució seria escalar K o c en funció de l'animació actual.

11.3 Treball futur

Calcular FE té una complexitat $O(nk)$ on n és el nombre de segments i $k = |Ch(K')|$. En aquest cas, com que K s'ha simplificat en un paral·lelogram, FE es calcula en $O(n)$. Per tant, el càlcul és apte per a realitzar-se cada *frame*. Una possible optimització consistiria a només recalculer la contribució de cada segment al FE quan superin un delta determinat.

Com s'ha mencionat a l'apartat anterior, amb aquest mètode poden existir falsos positius. Una millora futura consistiria a fer-lo més robust envers animacions extremes. Així i tot, en un sistema actiu de detecció de pèrdua de l'equilibri, sempre existirà la possibilitat que hi hagi falsos positius durant l'execució amb normalitat d'una animació; a causa de la naturalesa del problema.

²³Vegis apèndix A Codi 5 per a la implementació de l'algorisme per a calcular el FE

²⁴Aquelles animacions amb moviments d'alta freqüència com, per exemple, un *sprint*

12 Recuperació de l'equilibri

12.1 Definició del problema

En situació de desequilibri, les persones realitzem una resposta reflexa per tal de recuperar l'estat d'equilibri. Aquesta resposta de naturalesa biomecànica depèn de molts factors tals com la distància entre els peus o la percepció visual de l'entorn[48]. Entendre com la informació sensorial afecta a aquestes respostes reflexes és un tema que ha generat un interès considerable en el món acadèmic. Actualment no s'entén completament com els humans mantenen l'equilibri, especialment quan hi ha desordres que interfereixen en el procés[49][50][51].

Generalment, dins l'àrea de la fisiologia, es distingeixen tres tipus d'estratègies motores per a recuperar l'equilibri: les anomenades *ankle strategies*, *hip strategies* i *stepping strategies*. A la Figura 22 es pot apreciar aquestes estratègies des-de diferents angles. En les estratègies del turmell, la musculatura que envolta aquest ós s'ocupa d'exercir les forces necessàries per a mantenir l'equilibri de la resta del cós. Tot i que la selecció de les diferents estratègies depèn molt de l'edat i morfologia de l'individu, les *ankle strategies* se solen utilitzar en resposta a forces lleus i poc sobtades. Les *hip strategies*, en canvi, se solen utilitzar quan la força que produeix el desequilibri no es pot contrarestar amb la força exercida pels músculs dels turmells, però tampoc és necessari donar una passa. En aquest cas, la pelvis es mou de manera que el centre de massa es torni a reubicar sobre el polígon de suport. Finalment, les *stepping strategies*, s'activen quan la força rebuda no et pot compensar amb cap de les estratègies anteriors i cal donar una passa. La nova força de reacció amb el terra (GRF) permet desaccelerar el centre de massa i assolir un nou punt d'equilibri[52].

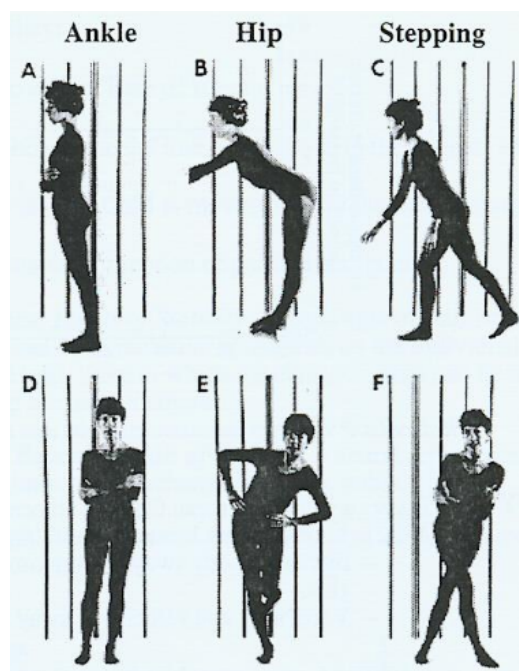


Figura 22: Tipus d'estratègies motores per a la recuperació de l'equilibri. (A,B,C) Mostren les *Ankle*, *Hip*, *Stepping strategies* respectivament, des d'un punt de vista lateral. (D,E,F) Mostren les mateixes estratègies frontalment.

Font: Faith Community Nurse Network

Així doncs, aquesta part del TFG busca trobar una forma de simular dites estratègies i alternar-les activament segons els esdeveniments externs. Com s'ha mencionat anteriorment, no es busca una simulació física fidel a la realitat. Es busca la generació de moviments, que siguin visualment convincents, mitjançant uns recursos computacionals bastant limitats.

12.2 Solució

12.2.1 Preliminars

En tots els sistemes tractats anteriorment, el *ragdoll* en cap moment s'ha simulat de forma completa. Amb el sistema que permet seguir moviments predefinitos, com s'ha mencionat a l'apartat 9, el moviment rectilini simulat del centre de massa s'ha desactivat. A l'apartat 10, on es tracta el sistema de cinemàtica inversa, la simulació del *ragdoll* s'havia desactivat per complet. El problema però, per a tractar un sistema de recuperació de l'equilibri convincent, cal simular totalment el *ragdoll*. Si es desactiva el moviment rectilini del centre de massa, per exemple, la resposta física envers impactes on s'hauria d'induir cert moment lineal al centre de massa serà molt antinatural.

Com ja s'ha justificat, l'inconvenient de simular el moviment rectilini del centre de massa és que les simulacions físiques d'un motor gràfic són molt aproximades. Per tant, no es pot confiar en elles perquè el *ragdoll* es mantingui dret de forma natural. L'única opció viable és realitzar una aproximació. En concret, es vol aproximar el que es coneix en el món de la biomecànica com a *ground reaction force* (GRF). Aquesta força, en línia amb la tercera llei de la mecànica Newtoniana, és la que manté el nostre centre de massa elevat i en equilibri quan ens trobem de peu[53].

L'aproximació que es proposa en aquest TFG consisteix a utilitzar de forma activa el que es coneix com a raycast en el motor *Unity3D*. Amb aquesta eina es pot detectar en tot moment si sota els peus del personatge hi ha una superfície sòlida. En cas que es detecti la superfície, s'aplica una força al centre de massa amb la mateixa direcció i diferent sentit a la de la gravetat. Aquesta força f_i que aplica cada punt de contacte²⁵ p_i es pot definir mitjançant la següent expressió proposada:

$$f_i = -\vec{g} * m * k * \cos(\hat{j}, \frac{\overrightarrow{CM} - \vec{p}_i}{\|\overrightarrow{CM} - \vec{p}_i\|})$$

On \vec{g} és el vector que descriu la força gravitacional, m la massa total del personatge, \overrightarrow{CM} el vector de posició del centre de massa, \vec{p}_i el vector de posició del punt de contacte i , \hat{j} el vector unitari vertical en coordenades globals i k el factor de contribució del punt de contacte. La idea és que k sigui la inversa del nombre de punts de contacte total. El model, però, es pot estendre perquè cada punt de contacte tingui una contribució diferent²⁶.

D'altra banda, s'ha mencionat la necessitat de fer que el sistema de cinemàtica inversa funcioni en conjunt amb les simulacions de *ragdoll*. En línia amb la solució definida a la secció 9,

²⁵En aquest TFG s'entén per a punt de contacte el punt en el qual una extremitat es troba col·lidint amb una superfície perpendicular a la gravetat

²⁶Per a simular personatges amb extremitats més fortes que altres, per exemple

només ha calgut aplicar les forces sobre les extremitats tenint com a orientacions objectiu les indicades per l'algorisme de cinemàtica inversa.

12.2.2 *Ankle strategy*

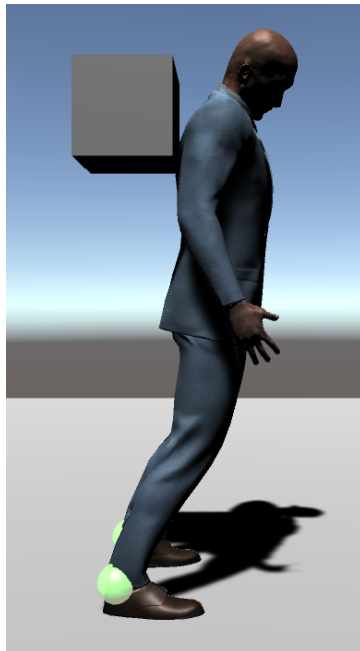


Figura 23: L'*ankle strategy* en acció. El cub aplica una força sobre la part posterior del tors que es veu compensada amb l'aparent força exercida pels músculs dels turmells.

Utilitzant el sistema de cinemàtica inversa a les extremitats inferiors, amb els objectius situats a la posició original dels peus, el *ragdoll* pot rebre impactes en la part superior del cos i tot i això seguir mantenint els peus en la mateixa posició. En cas contrari, els peus canviarien inevitablement de posició. Amb l'aproximació de GRF mencionada a l'apartat 12.2.1, el cos pot seguir de peu tot i estar inclinat. Ara bé, per tal de simular l'*ankle strategy* de forma natural, la part superior del *ragdoll* ha de tornar a la posició original de forma gradual.

Inicialment s'ha utilitzat el sistema desenvolupat a la secció 9 per a intentar mantenir en tot moment l'orientació inicial de la columna i el centre de massa. El problema, però, és que un impacte pot influir també en la posició del centre de massa. La solució proposada consisteix a estendre l'aproximació del GRF perquè s'apliqui una força addicional f_r al centre de massa. Aquesta força es pot definir mitjançant la següent expressió en un instant de temps $t = n$:

$$f_r(n) = k * \left(\frac{\overrightarrow{CM_{0e}} + \overrightarrow{CM_{0d}}}{2} - \overrightarrow{CM_n} \right)$$

on k és una constant per adaptar la velocitat amb la qual el centre de massa es pot moure, $\overrightarrow{CM_n}$ és la posició del centre de massa a l'instant de temps $t = n$, $\overrightarrow{CM_{0e}} = \vec{p}_{e0} - \overrightarrow{CM_0} + \vec{p}_{en}$ i \vec{p}_{en} és la posició del peu esquerre a l'instant de temps $t = n$. Anàlogament s'aplica la mateixa expressió per al peu dret. Per tant, f_r intentarà mantenir en tot moment la posició relativa original del centre de massa envers el peu dret i el peu esquerre.

12.2.3 Hip strategy

Dins l'àmbit de la recerca clínica, s'està d'acord en la definició d'aquesta estratègia utilitzant com a model un doble pèndol invertit (DIP) [54]. El model de pèndol invertit va ser introduït per primera vegada l'any 1972 per Vukobratović i Stepanenko [55]. Més tard va ser utilitzat en els àmbits de la robòtica i els gràfics [56] [57]. Des de llavors, s'ha realitzat moltes extensions del model a fi de simular diferents comportaments²⁷.

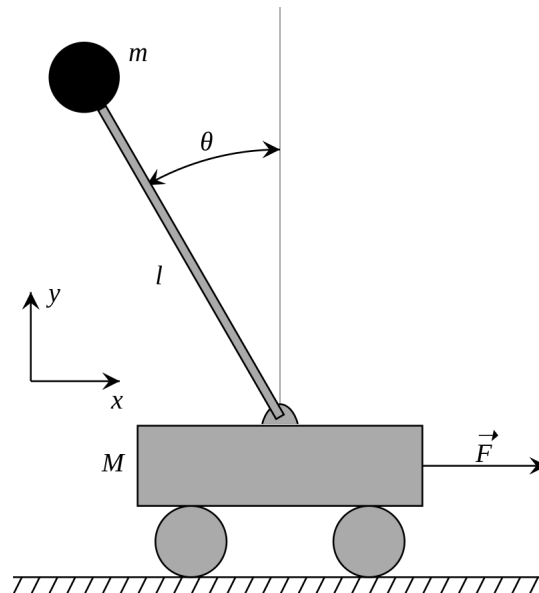


Figura 24: Esquema de l'IP

Generalment el pèndol invertit es defineix com un objecte de massa m lligada a un pal sense massa de longitud l que es troba sobre un objecte que es pot desplaçar com, per exemple, un carro de massa M . Generalment una unitat de control s'ocupa de dirigir els dos graus de llibertat del carro²⁸, utilitzant com a entrada la velocitat lineal de la massa lligada al pal o l'angle d'aquest respecte a l'eix vertical del carro, per mantenir el pal vertical. A la Figura 24 es mostra de forma esquemàtica el sistema. En les adaptacions per a modelar l'equilibri humà, el pal representaria les cames, l'objecte lligat al pal el centre de massa del cos i, el carro, els peus. Tot i que des d'un punt de vista estrictament energètic el sistema és consistent, el moviment resultant poc té a veure amb el que realitzem els humans.

El model del doble pèndol invertit consisteix a afegir una articulació al pal amb una segona massa. Aquesta articulació representaria el nou centre de massa i l'original el cap. Això produeix que el sistema resultant sigui més caòtic i requereixi un bon ajust dels diferents paràmetres [58]. Hwang [59] van aconseguir simular adequadament les estratègies utilitzant el model del pèndol invertit i un mètode basat en dades. El mètode proposat consistia a associar els diferents estats de la unitat de control que ha de mantenir el pèndol en equilibri amb un conjunt de possibles animacions predefinides utilitzant *Machine Learning*.

²⁷Per exemple, incorporar les lleis de Hooke per a simular el moviment oscil·latori de la pelvis al caminar

²⁸Assumint que es pot desplaçar sobre el pla $x - y$

Cap de les solucions proposades en aquest TFG utilitza un pèndol invertit. Primerament, perquè utilitzar directament un pèndol invertit no produeix moviments humanament realistes; encara que biomecànicament tinguin sentit. Utilitzar un pèndol invertit amb un mètode basat en dades si sembla una bona opció però, com ja s'ha justificat a la secció 2, en aquest TFG no es contempla aquesta possibilitat.

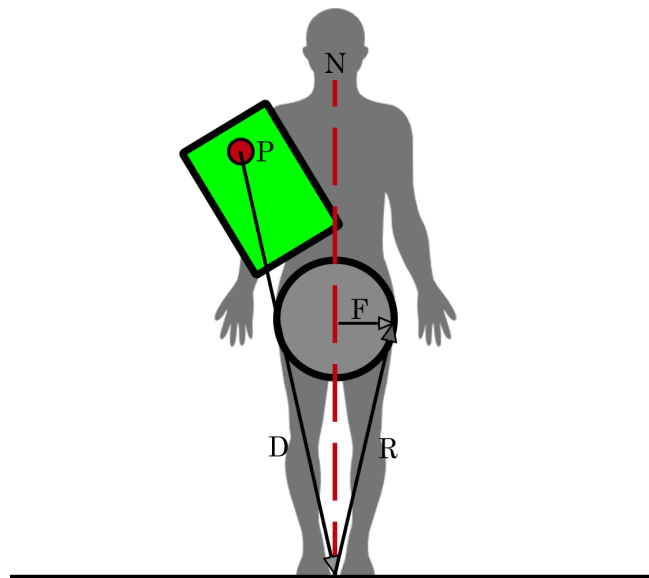


Figura 25: Esquema del mètode proposat per a simular una *hip strategy*

La solució proposada es basa en l'observació que en la *hip strategy* la pelvis sempre es mou en direcció oposada a la part superior del tors respecte a el CP²⁹. La idea principal és utilitzar un punt P de referència situat a la part superior del tors. A partir d'aquest punt, s'obté un vector $\vec{D} = CP - P$. És a dir, un vector amb direcció de P cap al centre de pressió. Si es reflecteix aquesta direcció envers la normal de la superfície de contacte, obtenim R . Amb R podem tenir una referència relativa al CP d'on hauria de moure's la pelvis per a compensar el pes del tors. Normalitzant R i escalant-la segons l'alçada de la pelvis s'obté aquest punt. Aplicant una força F des del punt d'origen de la pelvis fins al punt obtingut aconseguim que la pelvis es reposicioni de forma activa simulant l'estratègia desitjada. A la Figura 25 es mostra un esquema del mètode vist des del pla frontal. El cercle indica la pelvis, el rectangle verd la part superior del tors i N la normal de la superfície de contacte.

A la Figura 26 es mostra el resultat final. Com es pot veure, la part superior del tors del personatge rep un impacte per part del cub verd i l'impacte es veu compensat per un moviment dels malucs en direcció contrària.

²⁹ A la secció 12.2.4 s'ofereix una definició



(a) *Hip strategy* frontal



(b) *Hip strategy* lateral

Figura 26: *Hip strategy* al rebre un impacte en diferents direccions

12.2.4 Stepping strategy

Aquesta és l'última estratègia que ens pot mantenir en equilibri quan les anteriors no han estat suficients. La idea principal és que l'agent realitzi una passa on la nova GRF compensi els moments angulars i lineals del centre de massa.

El primer problema que cal resoldre és com generar el moviment d'una passa. Com que una passa es pot donar en diferents direccions i longituds, la quantitat de possibles passes en un instant determinat és molt elevada. Això fa que sigui inviable tenir un conjunt d'animacions predeterminades.

D'una manera similar a Wu i Zordan[60], es proposa un sistema per a generar passes de forma dinàmica utilitzant cinemàtica inversa. La idea consisteix a, donada una posició objectiu i l'*end-effector* d'un dels dos peus, interpolat l'*end-effector* des del peu fins a la posició objectiu, seguint una determinada trajectòria. Es pot tenir control sobre la trajectòria de l'*end-effector* utilitzant corbes de Bézier.

Les corbes de Bézier són un tipus de corba paramètrica on la seva forma pot esser modificada mitjançant n punts que reben el nom de "punts de control". Aquesta família de corbes es defineix mitjançant l'expressió:

$$B(t) = \sum_{k=0}^n B_{n,k}(t)P_k$$

Una corba Bézier de grau n necessita $n + 1$ punts de control per a ser definida. La corba resultant interpola P_0 i P_n i aproxima la resta de punts P_1, P_2, \dots, P_{n-1} . $B_{n,k}(t)$ són els *polinomis de Bernstein*. Es defineixen mitjançant:

$$B_{n,k} = \binom{n}{k} t^k (1-t)^{n-k}$$

També es poden definir amb la següent recurrència:

$$B_{n,k}(t) = (1-t)B_{n-1,k-1} + tB_{n-1,k}$$

La Figura 27 mostra un exemple de corba de Bézier de grau 3 i els seus corresponents polinomis base de *Bernstein*.

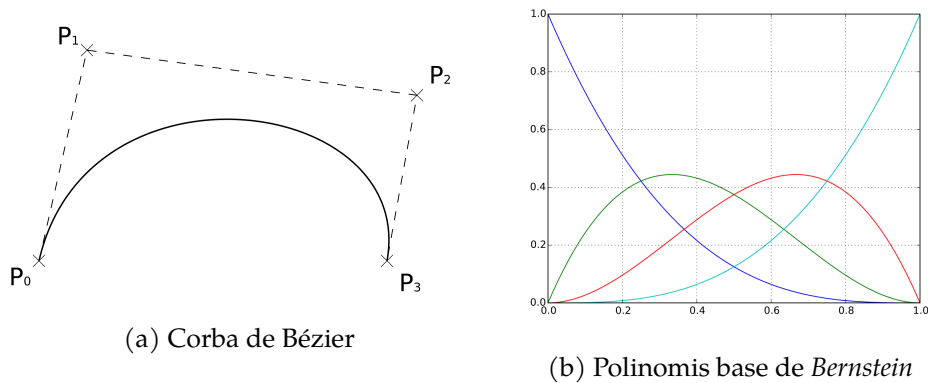


Figura 27: (a) Corba de Bézier de grau 3 definida pels punts P_0 , P_1 , P_2 i P_3 , (b) els polinomis base que la formen $B_{3,0}$ (blau), $B_{3,1}$ (verd), $B_{3,2}$ (vermell) i $B_{3,3}$ (cian) .

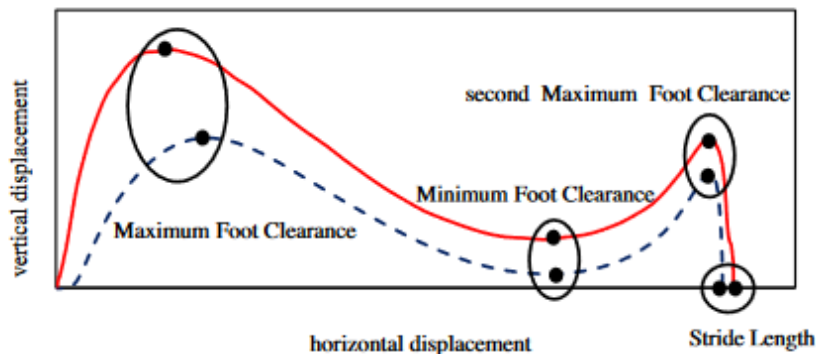
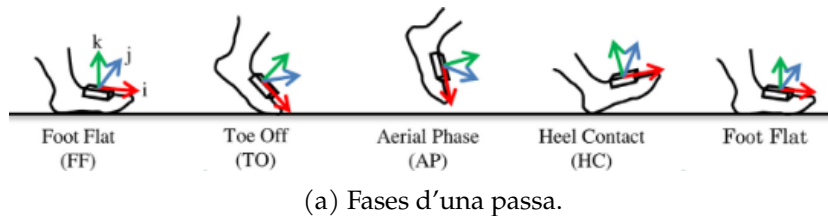


Figura 28: Les diferents fases d'una passa (a) i la trajectòria típica (b) que s'ha utilitzat com a referència per a l'aproximació.

Font: Elsevier

Per a construir la trajectòria d'una passa, s'ha utilitzat com a referència els resultats obtinguts per Kitagawa[61]. En aquest article, utilitzen un acceleròmetre i un giroscopi per a determinar la trajectòria de les passes en 10 adults sans. Els resultats (Figura 28), similars als que es podrien obtenir amb captura de moviment, ofereixen una visió comprensiva del comportament d'una passa.

Basant-se en els resultats de Kitagawa, s'ha conclòs que la fase d'elevació màxima d'una passa sol ocórrer aprop del 20% inicial del desplaçament total del peu. Per aproximar aquesta trajectòria s'ha utilitzat una corba de Bézier de grau 3:

$$B(t) = (1-t)^3O + 3(1-t)^2th + 3(1-t)t^2k + t^3T, 0 \leq t \leq 1$$

La Figura 29 mostra la corba resultant escalada a l'interval horitzontal $0 \leq t \leq 10$. O indica el punt inicial, T el punt de destí i h i k s'han ubicat de forma empírica; permeten modular l'alçada màxima de la passa i la forma general de la trajectòria respectivament.

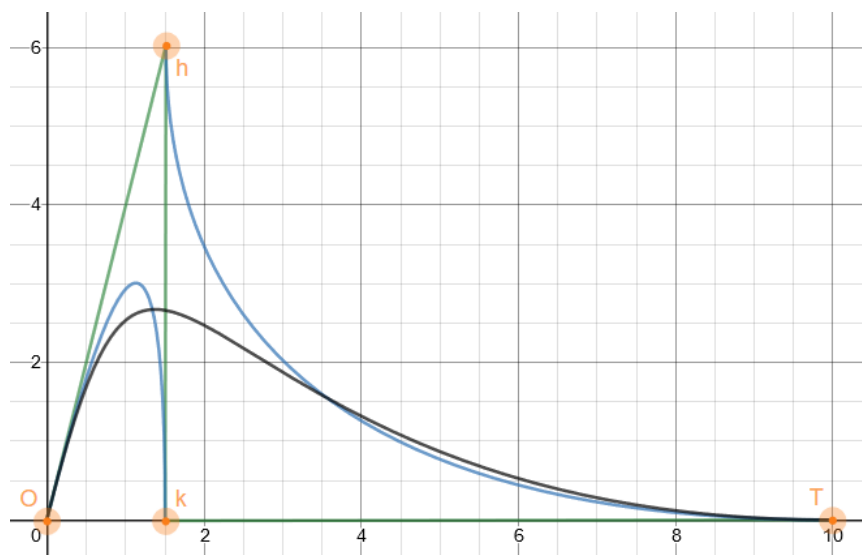


Figura 29: Trajectòria del peu aproximada amb una corba de Bézier.

Adicionalment i en línia amb Wu i Zordan[60], el centre de massa es desplaça al punt mig entre la nova posició del peu que ha donat la passa i el peu contrari. Per a tal efecte, s'utilitza una força constant; la magnitud de la qual es modula en funció de la velocitat desitjada de la passa.

Per cada fase com les mostrades a la Figura 28 s'ha emmagatzemat el quaternió corresponent a la rotació del peu indicada. Quan una passa es realitza, la rotació actual del peu es va interpolant esfèricament amb la de cadascuna de les fases en funció de d_i/d_t on d_i és la distància de l'*end effector* envers l'objectiu i d_t la distància total.

Un cop implementat el sistema de passes dinàmiques³⁰, ha calgut adaptar-lo de manera que les passes es dirigeixin de forma protectora en cas de desequilibri. Wu i Zordan[60] proposen un sistema basat en el centre de pressió CP. El centre de pressió, en l'àmbit de la biomecànica, és el punt d'aplicació de la GRF. Popvic i Goswami[62] mostren com el CP es pot expressar mitjançant el moment angular i lineal del centre de massa quan només es tenen en compte superfícies de suport horitzontals. Sigui \vec{c} el vector de posició del centre de massa, f la GRF i L i H els moments lineals i angulars del centre de massa respectivament, defineixen les components horitzontals del CP:

³⁰Vegis apèndix A Codi 3 per a la implementació de l'algorisme de passes dinàmiques i la corba de Bézier

$$p_x = c_x - \frac{L_x}{f_z} c_z - \frac{H_z}{f_z}$$

$$p_y = c_y - \frac{L_y}{f_z} c_z + \frac{H_x}{f_z}$$

Aquesta expressió prové de l'expansió de la relació entre el moment angular total del cos respecte al CP i el moment angular que la gravetat genera al cos respecte al CP, quan només es té en compte la força de la gravetat i la GRP: $M_{cp} = (\overrightarrow{CM} - \overrightarrow{CP}) \times M\vec{g}$.

Macchietto i Wu i Zordan[63][60], expliquen com un canvi desitjat per a contrarestar els actuals moments lineal i angular es pot expressar mitjançant:

$$L_{des} = -d_l L$$

$$H_{des} = -d_h H$$

On d_l i d_h són uns coeficients determinats per l'usuari. Si es substitueixen aquestes expressions en la de la posició del CP, podem obtenir l'expressió del CP desitjat³¹ per tal d'anul·lar els moments:

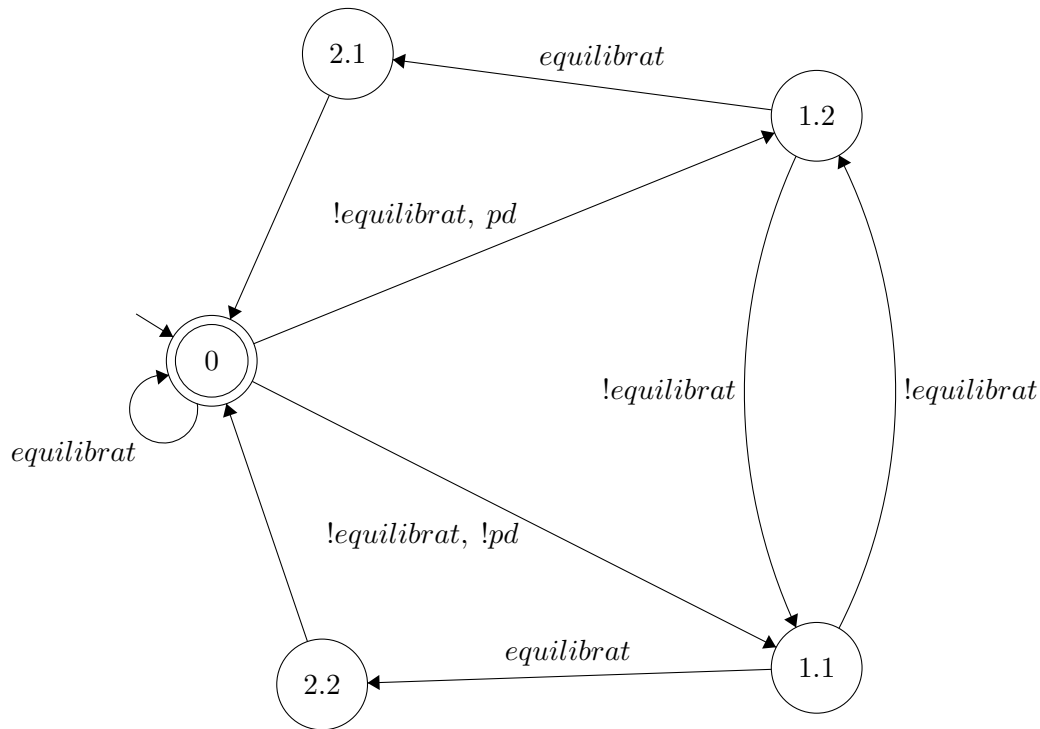
$$p_{xdes} = c_x + \frac{d_l L_x}{f_z} c_z + \frac{d_h H_z}{f_z}$$

$$p_{ydes} = c_y + \frac{d_l L_y}{f_z} c_z - \frac{d_h H_x}{f_z}$$

El criteri per a realitzar un pas protector és que el CP desitjat superi el polígon de suport. En aquest cas, es generarà una passa utilitzant l'expressió del CP desitjat amb els coeficients d_l i d_h augmentats per un cert factor. Aquest factor determinarà com de llarga serà la passa en funció de la pertorbació. d_l i d_h influiran en com de conservador serà l'agent. Valors alts faran que l'agent tingui una major tendència a donar una passa. Els valors $d_l = 4$ i $d_h = 6$ han donat bons resultats.

Wu i Zordan[60] proposen utilitzar el peu més proper al CP desitjat per a realitzar la passa. El problema és, que si cal més d'una passa per assolir l'estabilitat, es tornarà a escollir el mateix peu per a realitzar la nova passa. A més a més, es vol que un cop s'hagi assolit l'estabilitat de nou, els peus tornin a les seves respectives posicions originals per assolir la màxima estabilitat. Per aquest motiu, es proposa utilitzar una màquina d'estats finits (FSM) que gestioni quin peu donarà la pròxima passa. A continuació es mostra la FSM:

³¹Vegis apèndix A Codi 4 per a una implementació del seu càlcul



Per raons de simplicitat, no es tracta el cas en el qual l'agent pugui perdre l'equilibri mentre es troba reposicionant un dels peus. Cada estat s'ocupa de moure un dels *end-effectors*. Els estats $x.1$ mouen l'*end-effector* esquerre i els $x.2$ el dret. Els estats $1.x$ són els que tracten les passes de protecció i els $2.x$ les de reposicionament. L'estat 0 és el per defecte quan no s'ha de prendre cap acció. La FSM només s'avalua quan l'estat actual s'ha completat, un estat es completa si l'*end-effector* ja ha realitzat la interpolació. La condició *equilibrat* és certa si CP_{des} es troba a l'interior del polígon de suport, *pd* és cert si el peu dret és el més proper a CP_{des} . En les transicions on no apareix una o les dues variables, significa que els seus valors són intrascendents.

A la Figura 30 es mostra el resultat final. La seqüència de fotogrames mostra per ordre l'animació resultant en rebre un impacte per part del cub vermell. Com es pot veure, l'impacte resulta en el reposicionament del peu esquerre.

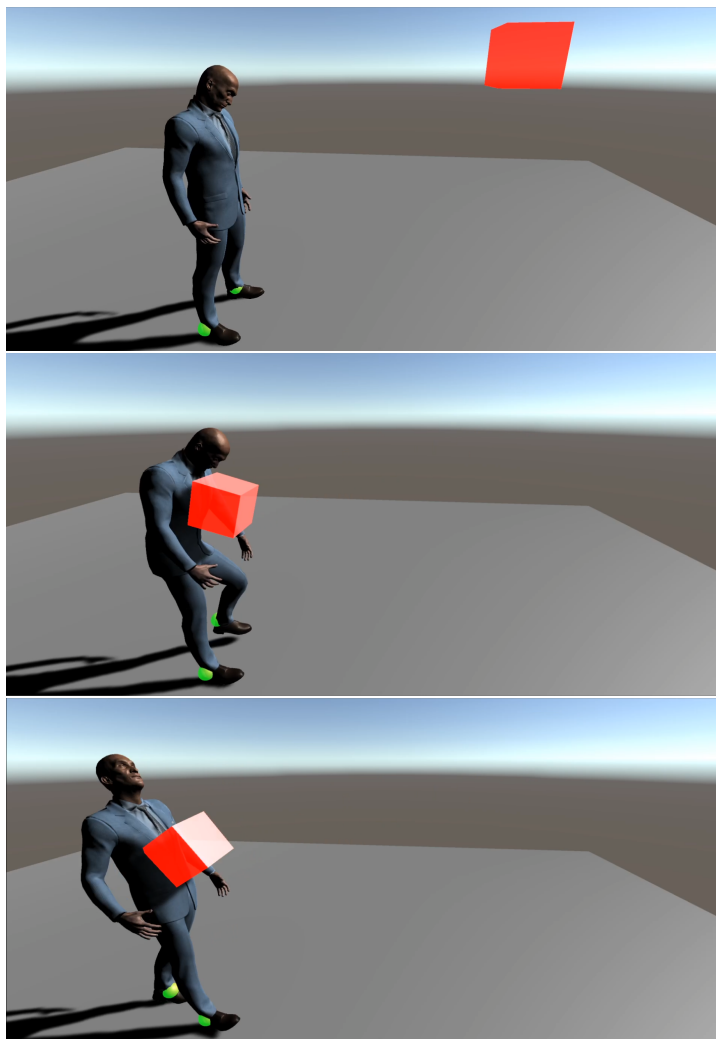


Figura 30: Seqüència de la *stepping strategy* simulada.

12.3 Treball futur

Els resultats obtinguts són millorables, com es pot apreciar a les Figures 23, 26 i 30. El fet de no utilitzar dades extretes per captura de moviment implica que el moviment resultant no serà tant fidel a la realitat com el generat amb un mètode basat en dades. Ara bé, els sistemes proposats permeten una major flexibilitat i adaptació a tota mena de casos d'ús. Aproximar el GRF ha demostrat ser una bona opció per oferir als *ragdolls* una major robustesa. El model d'aproximació, però, es podria ampliar per a tenir en compte altres factors tals com les forces de reacció laterals i la fricció de la superfície de contacte, per exemple. També es podria millorar la forma de detectar el contacte amb el terra, per simplicitat s'ha utilitzat un test d'intersecció en forma de raig, però es podria utilitzar una millor aproximació com, per exemple, un paral·lelepípedic.

Utilitzar el punt de contacte conjuntament amb el sistema desenvolupat a l'apartat 11 ha mostrat ser un bon mètode per alternar les diferents estratègies. De fet, la primera estratègia que es va implementar va ser la de *stepping*. Quan es va implementar l'estratègia de tipus *hip*, es

va veure com la tendència a donar una passa es reduïa de forma automàtica. El motiu era que aquesta última estratègia prevenia que el CP sortís del polígon de suport prematurament; com succeeix a la realitat. Per tant, no ha calgut implementar un sistema secundari que alterni entre les diferents estratègies.

El moviment resultant és funcional³² però caldria millorar la naturalitat amb què es produeix aquest moviment. Algunes extremitats es mouen per inèrcia a les forces aplicades sobre extremitats veïnes; caldria buscar algun mètode per a modular aquestes forces i obtenir uns moviments més consistents i suaus.

Ha quedat pendent estudiar el comportament dels braços en situació de desequilibri i crear controladors específics per a simular-lo. Finalment, caldria millorar la consistència d'aquest sistema per a casos més extrems, on intervenen forces de gran magnitud. En aquests casos, el comportament de l'algorisme és molt irregular.

Respecte al rendiment, utilitzant l'eina de *profiling* que ofereix Unity, s'ha pogut veure com l'impacte dels sistemes no supera els 4 mil·lisegons³³, tal com es pot apreciar a la Figura 31. En blau es mostra els temps d'execució del codi i en taronja els temps de còmput dels sistemes de físiques del motor; durant un interval de temps determinat. Per tant, la solució proposada en aquest TFG és totalment apte per una aplicació en temps real.

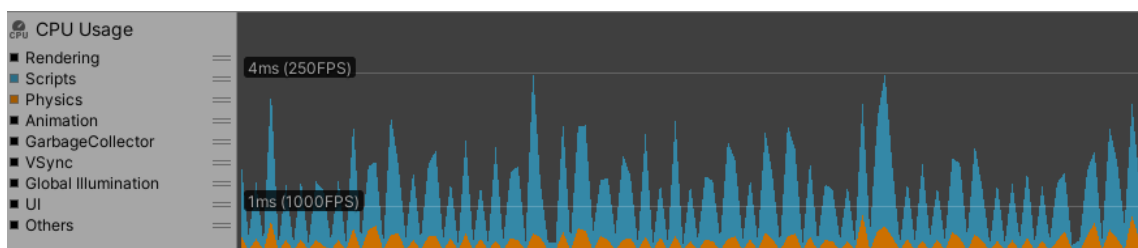


Figura 31: Gràfic del rendiment de l'algorisme de recuperació de l'equilibri.

³²En el sentit que les extremitats es mouen on cal, segons la situació en la qual es trobi el personatge

³³Executat amb un processador Intel i7-4790k

13 Conclusions finals

El potencial de les animacions dinàmiques és indubtable. El grau de dinamisme i flexibilitat que poden oferir a molts de projectes, dins l'àmbit d'aplicacions gràfiques en temps real, fa que sigui una opció molt atractiva. Com s'ha pogut comprovar en aquest TFG, la seva implementació no és trivial i requereix implementar diversos sistemes. Aquests sistemes, tot i semblar resoldre problemes inconnexos i pertànyer a diferents àrees d'estudi, en conjunt han estat clau per a resoldre un dels principals comportaments que es volia simular: la recuperació de l'equilibri.

Amb els *ragdolls* i un sistema que generi forces de torsió a cada articulació, com l'implementat a la secció 9, es pot generar qualsevol classe de moviment sense ignorar les restriccions de l'entorn on es produeixi.

Amb un sistema de cinemàtica inversa com el desenvolupat a l'apartat 10 podem manipular el moviment resultant del personatge des d'un alt nivell i fàcilment adaptar-lo al sistema de control de *ragdolls*. Algorismes heurístics per a sistemes multiarticulats com *FABRIK* no han acabat de funcionar del tot bé. En canvi, una combinació híbrida d'algorismes ha mostrat ser una bona manera d'afrontar el problema; complint les restriccions computacionals de les aplicacions en temps real.

Finalment, utilitzant els sistemes anteriors i certs coneixements de biomecànica, s'ha pogut simular una resposta de recuperació de l'equilibri. Durant l'estudi d'aquesta resposta s'ha determinat la seva complexitat en forma de diferents estratègies que treballen en sinergia. Desenvolupar diferents mecanismes de control, per a simular cadascuna d'aquestes estratègies, ha demostrat amb quina flexibilitat es pot incorporar nous comportaments actius a un *ragdoll* utilitzant únicament aproximacions basades en forces. Finalment, utilitzar corbes de Bézier per a guiar el moviment de les extremitats, tot i no ser un mètode novell, ha resultat ser un mètode simple, efectiu i flexible per a generar moviments basats en objectius; en el cas d'aquest TFG, ha estat el de fer passos.

Allunyar-se de qualsevol mètode basat en dades, tot i ser perfectament compatible amb l'objectiu final de generar moviments dinàmics, ha permès obtenir solucions fàcilment adaptables a tota mena de casos d'ús. De cara al futur seria interessant veure com pot funcionar un algorisme de *reinforcement learning*, conjuntament amb els sistemes desenvolupats, per a generar el mateix comportament de recuperació de l'equilibri i explorar altres de nous.

Per tant, es pot concloure que en aquest TFG s'ha aconseguit assolir els objectius principals establerts. És cert, però, que ha quedat pendent l'objectiu secundari d'implementar una altra resposta dinàmica; centrada en la prevenció d'impactes. A la meitat del desenvolupament d'aquest treball es va prendre la decisió de no implementar-la. El motiu ha estat poder realitzar un estudi amb més profunditat d'altres apartats com la cinemàtica inversa i la pèrdua d'equilibri, que tenen una major importància en l'objectiu principal d'aquest TFG. Tampoc es pot obviar el fet que cap de les solucions trobades en aquest treball és immillorable³⁴. Així i tot, s'espera que hagin mostrat una via de desenvolupament més flexible i oberta respecte a les alternatives existents.

³⁴Com es pot apreciar als subapartats 9.3 10.3 11.3 12.3

14 Bibliografia

Referències

- [1] T. Donovan, *Replay: The History of Video Games*. Yellow Ant, Abril 2010.
- [2] A. Beane, *3D Animation Essentials*. Sybex, Març 2012.
- [3] A. Aristidou, J. Lasenby, Y. Chrysanthou, and A. Shamir, "Inverse kinematics techniques in computer graphics: A survey," *Computer Graphics Forum*, vol. 37, no. 6, pp. 35–58, 2018.
- [4] J. Cooper, *Game Anim: Video Game Animation Explained*. A K Peters/CRC Press, Gener 2019.
- [5] ViRVIG, "ViRVIG official website," 2020. [Online]. Disponible: <https://www.virvig.eu/>.
- [6] T. Komura, I. Habibie, J. Schwarz, and D. Holden, *Data-Driven Character Animation Synthesis*, pp. 1–29. Cham: Springer International Publishing, 2017.
- [7] D. Holden, J. Saito, and T. Komura, "A deep learning framework for character motion synthesis and editing," *ACM Trans. Graph.*, vol. 35, July 2016.
- [8] CMU, "CMU Graphics Lab Motion Capture Database," 2020. [Online]. Disponible: <http://mocap.cs.cmu.edu/info.php>.
- [9] R. S. Sutton, *Introduction: The Challenge of Reinforcement Learning*, pp. 1–3. Boston, MA: Springer US, 1992.
- [10] NaturalMotion, "NaturalMotion official website," 2020. [Online]. Disponible: <https://www.naturalmotion.com/>.
- [11] D. Rosen, "Why we are not using euphoria," 2009. [Online]. Disponible: <http://blog.wolfire.com/2009/11/why-we-are-not-using-euphoria/>.
- [12] NaturalMotion, "Why isn't the euphoria engine used in more games?," Gener 2013. [Online]. Disponible: https://www.reddit.com/r/Games/comments/15wvoq/why_isnt_the_euphoria_engine_used_in_more_games/.
- [13] RootMotion, "RootMotion official website," 2020. [Online]. Disponible: <http://root-motion.com/>.
- [14] U. Technologies, "Root Motion | Unity Asset Store," 2020. [Online]. Disponible: <https://assetstore.unity.com/publishers/5300>.
- [15] A. Aristidou, D. Cohen-Or, J. Hodgins, and A. Shamir, "Self-similarity analysis for motion capture cleaning," *Computer Graphics Forum*, vol. 37, 05 2018.
- [16] S. Lee, M. Park, K. Lee, and J. Lee, "Scalable muscle-actuated human simulation and control," *ACM Trans. Graph.*, vol. 38, July 2019.
- [17] S. Starke, H. Zhang, T. Komura, and J. Saito, "Neural state machine for character-scene interactions," *ACM Transactions on Graphics*, vol. 38, 11 2019.

- [18] J. Kim, Y. Seol, H. Kim, and T. Kwon, "Interactive character posing with efficient collision handling," *Computer Animation and Virtual Worlds*, 03 2020.
- [19] K. Beck, M. Beedle, A. v. Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas, "The agile manifesto," 2001. [Online]. Disponible: <https://agilemanifesto.org/principles.html>.
- [20] FIB-UPC, "Normativa del treball final de grau del grau en enginyeria informàtica de la FIB," 2012. [Online]. Disponible: <https://www.fib.upc.edu/sites/fib/files/documents/actes/normativatfg-gei-2012-09-26.pdf>.
- [21] Payscale, "Average Recruit Salary in Spain," 2020. [Online]. Disponible: <https://www.payscale.com/research/ES/Job=Recruiter/Salary>.
- [22] Payscale, "Average Software Engineer Salary in Spain," 2020. [Online]. Disponible: https://www.payscale.com/research/ES/Job=Software_Engineer/Salary.
- [23] Payscale, "Average Project Manager, Information Technology (IT) Salary in Spain," 2020. [Online]. Disponible: [https://www.payscale.com/research/ES/Job=Information_Technology_\(IT\)_Manager/Salary](https://www.payscale.com/research/ES/Job=Information_Technology_(IT)_Manager/Salary).
- [24] Latex Project, "The latex project public license," 2008. [Online]. Disponible: <https://www.latex-project.org/lppl.txt>.
- [25] Atlassian, "Trello plans," 2020. [Online]. Disponible: <https://trello.com/en/pricing/>.
- [26] Unity Technologies, "Unity3d plans," 2020. [Online]. Disponible: <https://store.unity.com/compare-plans?currency=EUR>.
- [27] The Python Software Foundation, "Python: History and license," 2020. [Online]. Disponible: <https://docs.python.org/3/license.html>.
- [28] Adobe, "Mixamo faq," 2020. [Online]. Disponible: <https://helpx.adobe.com/creative-cloud/faq/mixamo-faq.html>.
- [29] Unity Technologies, "Asset store terms of service and eula," 2020. [Online]. Disponible: <https://helpx.adobe.com/creative-cloud/faq/mixamo-faq.html>.
- [30] 021BCN, "Pressupost espai co-working a poblenou," 2012. [Online]. Disponible: <http://www.021espai.com/tarifas-coworking-barcelona-poblenou/>.
- [31] U. Technologies, "Unity terms of service," 2020. [Online]. Disponible: <https://unity3d.com/legal/terms-of-service>.
- [32] Microsoft, "Rocketbox terms of service," 2020. [Online]. Disponible: <https://github.com/microsoft/Microsoft-Rocketbox/blob/master/LICENSE.md>.
- [33] C. C. Center, "About Copyright," 2020. [Online]. Disponible: <https://s100.copyright.com/AppDispatchServlet>.

- [34] E. B. Dam, M. Koch, and M. Lillholm, *Quaternions, interpolation and animation*, vol. 2. Cite-seer, 1998.
- [35] I. Millington, "12 - collision detection," in *Game Physics Engine Development (Second Edition)* (I. Millington, ed.), pp. 253 – 289, Boston: Morgan Kaufmann, second edition ed., 2010.
- [36] Wikipedia, "Quaternions and spatial rotation," 2020. [Online]. Disponible: https://en.wikipedia.org/wiki/Quaternions_and_spatial_rotation.
- [37] P. Wrotek, O. C. Jenkins, and M. McGuire, "Dynamo: Dynamic, data-driven character control with adjustable balance," in *Proceedings of the 2006 ACM SIGGRAPH Symposium on Videogames, Sandbox '06*, (New York, NY, USA), p. 61–70, Association for Computing Machinery, 2006.
- [38] A. Aristidou, Y. Chrysanthou, and J. Lasenby, "Extending fabrik with model constraints," *Computer Animation and Virtual Worlds*, vol. 27, pp. 35–57, 02 2016.
- [39] J. J. Craig, *Introduction to Robotics: Mechanics and Control*. USA: Addison-Wesley Longman Publishing Co., Inc., 2nd ed., 1989.
- [40] J. Q. Gan, E. Oyama, E. M. Rosales, and H. Hu, "A complete analytical solution to the inverse kinematics of the pioneer 2 robotic arm," *Robotica*, vol. 23, no. 1, pp. 123–129, 2005.
- [41] M. Raghavan and B. Roth, "Inverse Kinematics of the General 6R Manipulator and Related Linkages," *Journal of Mechanical Design*, vol. 115, pp. 502–508, 09 1993.
- [42] A. Aristidou, J. Lasenby, Y. Chrysanthou, and A. Shamir, "Inverse kinematics techniques in computer graphics: A survey," *Computer Graphics Forum*, vol. 37, pp. 35–58, 09 2018.
- [43] S. Buss, "Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods," *IEEE Transactions in Robotics and Automation*, vol. 17, 05 2004.
- [44] A. Aristidou and J. Lasenby, "Fabrik: A fast, iterative solver for the inverse kinematics problem," *Graph. Models*, vol. 73, p. 243–260, Sept. 2011.
- [45] S. J. Wright, "Coordinate descent algorithms," *Mathematical Programming*, vol. 151, no. 1, pp. 3–34, 2015.
- [46] J. Chai and J. K. Hodgins, "Performance animation from low-dimensional control signals," *ACM Trans. Graph.*, vol. 24, p. 686–696, July 2005.
- [47] R. McGhee and A. Frank, "On the stability properties of quadruped creeping gaits," *Mathematical Biosciences*, vol. 3, pp. 331 – 351, 1968.
- [48] B. Day, M. Steiger, P. Thompson, and C. Marsden, "Effect of vision and stance width on human body motion when standing: implications for afferent control of lateral sway," *The Journal of physiology*, vol. 469, p. 479—499, September 1993.
- [49] P. Faloutsos, M. van de Panne, and D. Terzopoulos, "The virtual stuntman: dynamic characters with a repertoire of autonomous motor skills," *Comput. Graph.*, vol. 25, pp. 933–953,

2001.

- [50] H. van der Kooij, B. Koopman, R. Jacobs, T. Mergner, and H. Grootenboer, "Quantification of sensory information in human balance control," vol. 5, pp. 2393–2396 vol.5, 1998.
- [51] R. C. Fitzpatrick, J. L. Taylor, and D. I. McCloskey, "Ankle stiffness of standing humans in response to imperceptible perturbation: reflex and task-dependent components," *The Journal of physiology*, Aug 1992.
- [52] P. Gatev, S. Thomas, T. Kepple, and M. Hallett, "Feedforward ankle strategy of balance during quiet stance in adults," *The Journal of Physiology*, vol. 514, pp. 915 – 928, 09 2004.
- [53] D. A. Winter, "Kinematic and kinetic patterns in human gait: Variability and compensating effects," *Human Movement Science*, vol. 3, no. 1, pp. 51 – 76, 1984.
- [54] C. Runge, C. Shupert, F. Horak, and F. Zajac, "Ankle and hip postural strategies defined by joint torques," *Gait & Posture*, vol. 10, no. 2, pp. 161 – 170, 1999.
- [55] M. Vukobratović and J. Stepanenko, "On the stability of anthropomorphic systems," *Mathematical Biosciences*, vol. 15, no. 1, pp. 1 – 37, 1972.
- [56] S. Kajita and K. Tani, "Experimental study of biped dynamic walking in the linear inverted pendulum mode," in *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2885–2891 vol.3, 1995.
- [57] Shunsuke Kudoh, T. Komura, and K. Ikeuchi, "Stepping motion for a human-like character to maintain balance against large perturbations," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pp. 2661–2666, 2006.
- [58] A. Kot and A. Nawrocka, "Double inverted pendulum for human hip strategy of balance modelling," in *2016 17th International Carpathian Control Conference (ICCC)*, pp. 368–371, 2016.
- [59] J. Hwang, I. H. Suh, G. Park, and T. Kwon, "Human character balancing motion generation based on a double inverted pendulum model," in *Proceedings of the Tenth International Conference on Motion in Games, MIG '17, (New York, NY, USA)*, Association for Computing Machinery, 2017.
- [60] C.-C. Wu and V. Zordan, "Goal-directed stepping with momentum control," in *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '10, (Goslar, DEU)*, p. 113–118, Eurographics Association, 2010.
- [61] N. Kitagawa and N. Ogihara, "Estimation of foot trajectory during human walking by a wearable inertial measurement unit mounted to the foot," *Gait & Posture*, vol. 45, 01 2016.
- [62] M. B. Popovic, A. Goswami, and H. Herr, "Ground reference points in legged locomotion: Definitions, biological trajectories and control implications," *The International Journal of Robotics Research*, vol. 24, no. 12, pp. 1013–1032, 2005.
- [63] A. Macchietto, V. Zordan, and C. Shelton, "Momentum control for balance," *ACM Trans. Graph.*, vol. 28, 07 2009.

Glossari

captura de moviment Procés que consisteix a gravar el moviment d'objectes reals i traslladar-lo a objectes digitals. 7, 9, 11–13, 40, 59, 63

centre de massa Punt d'un cos on l'acumulació relativa de tota la seva massa respecte a aquest punt és zero. 33, 36, 48, 53–55, 58, 60

convex hull Donada una figura geomètrica, el *convex hull* d'aquesta figura és el mínim conjunt de punts convex que la conté. 49

corba paramètrica Conjunt d'equacions que defineixen una corba en funció d'una variable t . 58

deep learning Conjunt d'algorismes d'aprenentatge automàtic que intenta generar abstraccions d'alt nivell en les dades. 13

FSM Model de computació abstracte que, en un moment determinat, només es pot trobar en un estat concret. 61, 62

GRF Conjunt de forces tant musculars com gravitacionals durant l'estada de peu. 52–54, 58, 60, 63

motion graph Graf dirigit on els nodes són seqüències d'animació i , les arestes, transicions entre les diferents seqüències. En un moment determinat, el personatge només pot estar reproduint l'animació d'un determinat node i , segons unes condicions definides per l'usuari a cada aresta, passar d'un node a un altre. 14

polígon de suport Regió horitzontal sobre la qual el centre de massa d'un objecte ha de pertànyer per assolir l'estabilitat. 48–50, 52, 61, 62, 64

quaternions Sistema numèric resultat d'una extensió dels nombres complexos. Aquests en concret tenen dues components imaginàries addicionals j i k . Han resultat d'especial utilitat dins l'àmbit de gràfics per computador, ja que presenten una forma compacta per emmagatzemar rotacions i realitzar operacions. 34

raycast Molts motors de físiques ofereixen la possibilitat de realitzar tests d'intersecció. Aquests tests consisteixen a projectar una figura geomètrica a l'espai tridimensional i comprovar si hi ha alguna intersecció amb un o més objectes de l'escena. Els *Raycast*, en concret, són segments que es poden projectar en l'origen, direcció i magnitud que l'usuari desitgi. 53

test d'orientació Dins el camp de la geometria computacional, els tests d'orientació són un mètode per a determinar si un punt es troba a l'esquerra o a la dreta d'una línia, utilitzant el producte vectorial. Aquest mètode es pot estendre per a determinar interseccions entre segments, punts dins de polígons, etc. 49

A Codi font

En aquest apèndix es mostra el codi dels algorismes més importants que s'ha desenvolupat durant aquest TFG, la resta del codi es troba a <https://github.com/ortiiixx/TFG>.

Codi 1: Implementació del bucle principal de *FABRIK*

```
void BackwardReach()
{
    Joints[0].transform.position = InitialRootLocation;
    for (int i = 1; i < Joints.Count; i++)
    {
        FABRIKJoint Fab = Joints[i - 1].GetComponent<FABRIKJoint>();
        Joints[i].transform.parent = null;
        switch (Fab.RestrictionType)
        {
            case FABRIKJoint.JointType.Hinge:
                HingeConstraint(Joints[i - 1], Joints[i]);
                break;
            case FABRIKJoint.JointType.Socketball:
                SocketBallConstraint(Joints[i - 1], Joints[i], false);
                break;
        }
        Joints[i].transform.position = Joints[i - 1].transform.position +
        Joints[i - 1].rotation * Fab.Forward * Lengths[i - 1];
        ReorientateJoint(Joints[i - 1].transform, Joints[i].transform);
        Joints[i].transform.position = Joints[i - 1].transform.position +
        Joints[i - 1].rotation * Fab.Forward * Lengths[i - 1];

        Joints[i].transform.parent = Joints[i - 1].transform;
    }
}

void ForwardReach()
{
    Joints[Joints.Count - 1].transform.position = Target;
    for (int i = Joints.Count - 1; i > 0; i--)
    {
        FABRIKJoint Fab = Joints[i - 1].GetComponent<FABRIKJoint>();
        float r = Lengths[i - 1];
        float d = Vector3.Distance(Joints[i - 1].transform.position,
        Joints[i].transform.position);
        float lambda = r / d;

        Joints[i].transform.parent = null;
        Joints[i - 1].transform.position =
        Joints[i - 1].transform.position * lambda +
        Joints[i].transform.position * (1f - lambda);
        ReorientateJoint(Joints[i - 1], Joints[i]);
    }
}
```

```

        Joints[i].transform.position = Joints[i - 1].transform.position +
        Joints[i - 1].rotation * Fab.Forward * Lengths[i - 1];
        Joints[i].transform.parent = Joints[i - 1];
    }
}

// Run at each physics simulation step
void FixedUpdate()
{
    parentV = Joints[0].up;
    quatero = Joints[0].rotation;
    Target = (UpperObjectives[0].position +
    UpperObjectives[1].position) / 2f + OffsetUp;
    TargetLastPosition = Target;
    if (Vector3.Distance(InitialRootLocation, Target) > TotalLength)
    {
        Debug.Log("Target is unreachable.");
        Joints[0].transform.position = InitialRootLocation;
        for (int i = 1; i < Joints.Count; i++)
        {
            float r = Vector3.Distance(Joints[i].transform.position,
            Joints[i - 1].transform.position);
            float d = Vector3.Distance(Joints[i - 1].transform.position,
            Target);
            float lambda = r / d;
            FABRIKJoint FJ = Joints[i - 1].GetComponent<FABRIKJoint>();
            Joints[i].transform.position =
            Joints[i - 1].transform.position *
            (1f - lambda) + Target * lambda;
            ReorientateJoint(Joints[i - 1].transform, Joints[i].transform);
            Joints[i].transform.position =
            Joints[i - 1].transform.position +
            Joints[i - 1].rotation * FJ.Forward * Lengths[i - 1];
        }
    }
    else
    {
        Debug.Log("Target is reachable.");
        for (int c = 0; c < MaxIterations &&
        Vector3.Distance(Joints[Joints.Count - 1].transform.position,
        Target) > Threshold; c++)
        {
            ForwardReach(); //FORWARD PASS
            BackwardReach(); //BACKWARD PASS
        }
    }
}

```

Codi 2: Implementació de les restriccions hinge i socket-ball de *FABRIK*

```

void HingeConstraint(Transform Hinge, Transform Child)
{
    FABRIKJoint FJ = Hinge.GetComponent<FABRIKJoint>();
    Vector3 GlobalTwistAxis = FJ.transform.rotation * FJ.TwistAxis;
    Vector3 ProjectedPosition1 =
    Vector3.ProjectOnPlane(Child.position, GlobalTwistAxis);
    Vector3 ProjectedPosition2 =
    Vector3.ProjectOnPlane(Hinge.position, GlobalTwistAxis);
    Vector3 RelativePosition = ProjectedPosition1 - ProjectedPosition2;
    Child.position = Hinge.position + RelativePosition;
}

void SocketBallConstraint(
    Transform SocketBall,
    Transform Joint,
    bool forward)
{
    FABRIKJoint FJ = SocketBall.GetComponent<FABRIKJoint>();

    Vector3 LocalPos = Joint.transform.position -
    SocketBall.transform.position;
    Vector3 LocalUp = SocketBall.rotation * FJ.TwistAxis;
    Vector3 LocalRight = SocketBall.rotation * FJ.Axis;
    Vector3 LocalFwd = forward ? -Vector3.Cross(LocalUp, LocalRight) :
    Vector3.Cross(LocalUp, LocalRight);

    Vector3 newPos = Vector3.Project(LocalPos, LocalFwd);
    Vector3 O = SocketBall.transform.position + newPos;

    float l1 = newPos.magnitude * Mathf.Tan(Mathf.Deg2Rad * FJ.angle1);
    float l2 = newPos.magnitude * Mathf.Tan(Mathf.Deg2Rad * FJ.angle2);
    float l3 = newPos.magnitude * Mathf.Tan(Mathf.Deg2Rad * FJ.angle3);
    float l4 = newPos.magnitude * Mathf.Tan(Mathf.Deg2Rad * FJ.angle4);

    Vector3 Pos = Vector3.zero;
    Vector3 dir = Joint.transform.position -
    SocketBall.transform.position;

    Pos.x = Vector3.Dot(-LocalUp, dir);
    Pos.z = Vector3.Dot(LocalRight, dir);
    Joint.transform.position = Pos;
    Vector2 pos = GetQuadrantPosition(
    Joint.transform.position, l1, l2, l3, l4);
    Vector3 result = O;
    result -= LocalUp * pos.x;
    result += LocalRight * pos.y;
    Joint.transform.position = result;
}

```

```
}  
  
Vector2 GetQuadrantPosition(  
Vector3 Orpos,  
float l1,  
float l2,  
float l3,  
float l4)  
{ //Quadrants counter-clockwise  
  Vector2 Pos = new Vector2(Orpos.x, Orpos.z);  
  if (Pos.x <= l3 &&  
      Pos.x >= -l4 &&  
      Pos.y >= -l1 &&  
      Pos.y <= l2) //Within bounds  
    return Pos;  
  if (Orpos.x > 0 && Orpos.z > 0)  
  {  
    return GetClosestPoint(Pos, l2, l3);  
  }  
  else if (Orpos.z > 0) //cyan-red  
  {  
    return GetClosestPoint(Pos, l1, l3);  
  }  
  else if (Orpos.x < 0)  
  {  
    Pos.y *= -1;  
    Pos = GetClosestPoint(Pos, l1, l4);  
    Pos.y *= -1;  
    return Pos;  
  }  
  else  
  {  
    Pos.y *= -1;  
    Pos = GetClosestPoint(Pos, l2, l4);  
    Pos.y *= -1;  
    return Pos;  
  }  
}
```

Codi 3: Implementació de les passes dinàmiques i la corba de Bézier

```

IEnumerator PerformStep(bool isRightFoot)
{
    Vector3 velo = CM.velocity;
    float ControlPointHeight = .4f;
    float SimulationTime = .0f;
    Transform Foot = isRightFoot ? RightFoot : LeftFoot;
    Transform Effector = isRightFoot ? RightEffector : LeftEffector;
    Vector3 MidPoint = isRightFoot ? LeftFoot.position : RightFoot.position;
    //We want the midpoint between the other foot and the destination
    Vector3 InitialFootLocation = Foot.position;
    Vector3 FootControlPoint1 = (DesiredStep.position - Foot.position) / 5;
    FootControlPoint1 += Foot.position; FootControlPoint1 += Vector3.up *
    ControlPointHeight;
    Vector3 FootControlPoint2 = FootControlPoint1; FootControlPoint2.y =
    Foot.position.y;
    Vector3 vec = DesiredStep.position;
    Quaternion AerialPhase = isRightFoot ? RightFootAerial : LeftFootAerial;
    Quaternion GroundPhase = isRightFoot ? OriginalRightFootRotation :
    OriginalLeftFootRotation;

    MidPoint = (MidPoint + DesiredStep.position) / 2f;
    MidPoint.y += (transform.position - MidPoint).y;
    CanMoveFoot = false;
    vec.y = Effector.position.y;
    DesiredStep.position = vec;

    while (Vector3.Distance(Effector.position, DesiredStep.position) >
    effectorDistanceEpsilon)
    {
        float t = Vector3.Distance(Foot.position, DesiredStep.position) /
        Vector3.Distance(InitialFootLocation, DesiredStep.position);
        if (t < 0.5f)
            Foot.rotation = Quaternion.Slerp(GroundPhase, AerialPhase, t);
        else
            Foot.rotation = Quaternion.Slerp(AerialPhase, GroundPhase,
    t);

        if (Vector3.Distance(Effector.position, DesiredStep.position) <
    effectorDistanceEpsilon)
        {
            Effector.position = DesiredStep.position;
        }
        else
        {
            Vector3 pos = Bezier3(InitialFootLocation,
            FootControlPoint1,

```

```
        FootControlPoint2 , DesiredStep.position ,
        SimulationTime*MaxStepSpeed );
        Effector.position = pos;
    }

    CM.AddForce((MidPoint - transform.position) *
    CMControlForce*MaxStepSpeed*2.1f ,
    ForceMode.Acceleration );
    Debug.DrawLine(FootControlPoint1 ,
    (Foot.position + DesiredStep.position)
    / 2f , Color.red );
    Debug.DrawLine(transform.position , MidPoint);
    SimulationTime += Time.deltaTime;
    yield return null;
}

vec = Effector.position;
vec.y = DesiredStep.position.y;
Effector.position = vec;
Foot.rotation = GroundPhase;
yield return new WaitForSecondsRealtime(0.2f);
CanMoveFoot = true;
lastStepRight = isRightFoot;
}

public static Vector3 Bezier3(Vector3 s, Vector3 p1,
Vector3 p2, Vector3 e, float t)
{
    t = Mathf.Clamp(t, .0f, 1.0f);
    float rt = 1 - t;
    return rt * rt * rt * s + 3 * rt * rt * t * p1 +
    3 * rt * t * t * p2 + t * t * t * e;
}
```

Codi 4: Càlcul del SP

```
Vector3 GetDesiredPS(float dl, float dh)
{
    Vector3 res = Vector3.zero;
    if (CM){
        Vector3 angularMomentum = new Vector3(
            CM.inertiaTensor.x * CM.angularVelocity.x,
            CM.inertiaTensor.y * CM.angularVelocity.y,
            CM.inertiaTensor.z * CM.angularVelocity.z
        );
        res.y = LeftEffector.position.y;
        res.x = CM.position.x + CM.position.y *
            ((dl * CM.velocity.x * mass) / GroundReactionForce())
            + ((dh * angularMomentum.z) / GroundReactionForce());
        res.z = CM.position.z + CM.position.y *
            ((dl * CM.velocity.z * mass) / GroundReactionForce()) -
            ((dh * angularMomentum.x) / GroundReactionForce());
    }
    return res;}

```


Codi 5: Càlcul del FE

```

public float GetBalanceFactor()
{
    float counterPos = 0;
    float counterTot = 0;
    Vector3 p1 = Vector3.ProjectOnPlane(
    LeftFoot.transform.position +
    LeftFoot.transform.up * FootVerticalOffset -
    LeftFoot.transform.forward *
    FootHorizontalOffset, Vector3.up);

    Vector3 p2 = Vector3.ProjectOnPlane(
    RightFoot.transform.position -
    RightFoot.transform.up *
    FootVerticalOffset + LeftFoot.transform.forward *
    FootHorizontalOffset, Vector3.up);

    Vector3 p3 = Vector3.ProjectOnPlane(p2 +
    RightFoot.transform.up * FootLength, Vector3.up);

    Vector3 p4 = Vector3.ProjectOnPlane(p1 -
    LeftFoot.transform.up * FootLength, Vector3.up);

    List<float> buckets = new List<float>(){
    0.0f, 0.0f, 0.0f, 0.0f
    };

    foreach (KeyValuePair<Transform, BodyPart> k in
    BodyMapper.ToList())
    {
        FABRIKJoint FB = k.Key.GetComponent<FABRIKJoint>();
        Collider c = k.Key.GetComponent<Collider>();
        Rigidbody rb = k.Key.GetComponent<Rigidbody>();
        if (FB && c && rb)
        {
            counterTot += rb.mass;
            Vector3 bp1 = Vector3.ProjectOnPlane(
            FB.transform.position, Vector3.up);

            Vector3 bp2 = Vector3.ProjectOnPlane(
            FB.transform.position + FB.transform.rotation *
            FB.Forward * c.bounds.size.y, Vector3.up);

            Debug.DrawLine(FB.transform.position,
            FB.transform.position + FB.transform.rotation *
            FB.Forward * c.bounds.size.y, Color.green);
            Debug.DrawLine(Vector3.ProjectOnPlane(FB.transform.position,
            Vector3.up), Vector3.ProjectOnPlane(FB.transform.position +

```

```
FB.transform.rotation * FB.Forward *
c.bounds.size.y, Vector3.up), Color.red);
bool b1 =
MathHelpers.CheckPointInQuad(bp1, p1, p2, p3, p4);
bool b2 =
MathHelpers.CheckPointInQuad(bp2, p1, p2, p3, p4);
if (b1 && b2)
{
    counterPos += rb.mass;
}
else if (b1 != b2)
{
    float r = CheckPointRatio(bp1, bp2, p1, p2, p3, p4);
    counterPos += rb.mass*r;
    buckets[GetBucket(bp1, bp2)] += rb.mass*(1.0f-r);
}
else
{
    buckets[GetBucket(bp1, bp2)] += rb.mass;
}
}
float t = 2.0f*Mathf.Min(buckets[0],buckets[3]);
t += 2.0f*Mathf.Min(buckets[1],buckets[2]);
counterPos += t;
float balanceFactor = counterPos / counterTot;
return balanceFactor;
}
```