Universidad de Murcia

Facultad de Informática

Robust Feedback Control of Omnidirectional Biped Gait on the Nao Humanoid Robot

Control Retroalimentado Robusto de Marcha Bípeda Omnidireccional en el Robot Humanoide Nao

Juan José Alcaraz Jiménez

2014

A Juan, Eli y Lourdes

Acknowledgements

Debo comenzar por agradecer la ayuda de mis directores de tesis: los Profesores Humberto Martínez Barberá y David Herrero Pérez. Esta tesis parte de la experiencia y los desarrollos llevados a cabo por ambos durante años. Además, la confianza y las concisas directrices de Humberto, así como las correcciones aportadas por David, han propulsado y mejorado este trabajo.

Por otro lado, esta tesis ha sido posible gracias a la financiación del Ministerio de Educación, Cultura y Deporte mediante la Beca AP2008-01816 del Programa de Formación del Profesorado Universitario.

I would like to thank Professor Sven Behnke for receiving me at the AIS group in Bonn and for his help during my internship. There I met Marcell Misura, who generously shared his experience in balance control of biped robots. Their advises and remarks contributed meaningfully to the quality of the subsequent publications.

Je voudrais aussi rémercier le Professeur Pierre Blazevic pour l'oportunité de travailler au LISV sous la direction du Professeur Vincent Hugel. L'enthousiasme, l'attention et l'experience de Vincent ont fait de la participation à la RoboCup une experience agréable et enrichissante.

Del mismo modo, nunca podré olvidar las noches de programación compartidas con mis compañeros del equipo *Hidalgos*. Entre palomitas, naocidios, dulcinantes, crazies y ricotazos disfruté de vuestro conocimiento y amistad.

También estoy muy agradecido a los amigos y compañeros que he encontrado desde Cartagena hasta París, pasando por Murcia, Rennes y Bonn. Principalmente a aquellos con quienes he compartido cada avance y cada percance de este trabajo, ¡aunque también a los que me han hecho olvidarlo!

Y sobre todo, quiero reconocer el ilimitado apoyo y cariño de mi familia. Nunca podré agradecer como se merecen vuestros consejos, atenciones y alegrías.

Abstract

In order to allow humanoid robots to collaborate with humans and free them from dangerous or unpleasant activities, they must be able to perform reliable gaits. In last decades, the balance of humanoid robots has attracted an increasing attention of the scientific community. However, even though important results have been achieved in the theoretical level and in simulations, the balance control of physical robots still remains an open issue. Since humanoid robots are intended to operate in everyday environments, it is fundamental to validate balance control approaches in physical platforms and to face them to stressing conditions. In this work we propose a balance control system that improves walking robustness and disturbance rejection. The approach combines the zero moment point stability criterion with the control of angular momentum and step timing.

The robotic platform employed to validate these developments is the 2010 version of the commercial humanoid robot Nao, developed by the French company Aldebaran Robotics. This robot is 57 *cm* tall and accounts for 21 degrees of freedom, of which 11 are in the legs. The study of the special features of the platform is a key prerequisite for the design of the balance control approach. Therefore, we thoroughly review the technical specifications of the humanoid robot Nao and analyze the performance of its sensors and actuators. This analysis permits to determine the sensory-motor lag and the level of backslash at the torso. Moreover it reveals the incapacity of force sensing resistors to produce an usable estimate of the position of the center of pressure. To complete the analysis of the platform, we review the calculations required to solve the forward kinematics problem and propose a solution to the inverse kinematics problem. The innovative configuration of the legs of Nao, with two coupled yaw-pitch joint actuators at the hip, requires either the use of a computationally intensive solver or the development of customized inverse kinematics approach. We have chosen to develop a solution consisting of three stages. The first two stages use p-controllers to prepare the inputs to the third stage, which is solved analytically. The resulting approach is both computationally efficient and accurate.

Subsequently, we describe the modules required to generate the walking pattern of the gait, that is, the sequence of poses for the robot that will permit it to attain a certain target location or velocity. Through the use of the aforementioned inverse kinematics module it is possible to specify these poses in the convenient Cartesian space. The poses of the robot are therefore described by means of the configuration of the pose of the feet, the location of the center of mass (CoM) and the orientation of the torso, which is kept vertical.

The first stage to generate the walking pattern is to derive the location of the footsteps from the motion type requested. Finding the optimum location for a step is a complex problem that has not yet been solved. Hence, many developments employ empirical approaches to generate the footstep locations. In this work we use a combination of heuristic and analytic criteria to obtain a solution to the problem. The optimal footstep location depends on several factors. The first of these factors is related to the linear and rotational inertia of the robot: if abrupt changes in the trajectory of the motion happen, the state of the system may exceed the capacity of the balance control algorithms to stabilize the robot. Another factor to be considered in the planning of the footsteps is the range of the legs. The maximum size of a step depends on the configuration of the kinematic chain of the robot. We will take into account these constraints to generate the sequence of future footsteps.

Once computed the target footstep location, we will focus on the trajectory of the stride that will take the swing foot to its destination. The trajectories of the swing foot can induce a tipping moment on the robot due to the reaction forces generated by the collisions against the ground or to the inertial forces that appear when the foot is accelerated. Moreover, high inertial forces also prevent the servomotors from effectively tracking the commanded reference positions. In this work, the trajectories of the soles of the feet are always kept parallel to each other and are defined by piecewise cubic Bézier spline curves. The design of the trajectories aims at preventing the impacts of the swing foot against the ground that are likely to occur at the heel and the toes. Additionally, this approach allows online modification of the landing position and provides flexibility to generate complex trajectories.

Another paramount contribution to the balance of walking robots arises from the movement of the CoM. Although the design of trajectories for the CoM remains open to discussion, we can identify

the use of the Zero Moment Point (ZMP) stability criterion applied to an inverted pendulum model as a common element to most of the locomotion systems. This criterion states that the stability of the robot at every instant is conditioned by the existence of the ZMP within the support polygon of the robot. Thus, to generate stable walking patterns for the robot at a certain instant, it is sufficient to generate trajectories for the CoM that allow to project the ZMP within the support polygon at that instant and in the future. To solve this problem we will use a type of controller that provides an optimal solution for a limited future interval: the Model Predictive Controller (MPC). This controller relies on a model of the system dynamics and minimizes a cost function over the receding prediction horizon. In this way, the MPC allows to optimize the control action at an instant, taking into consideration the future evolution of the system. Since the MPC offers its best performance when applied to linear models, the inverted pendulum model cannot be directly employed. The controller uses instead the Three Dimensional Linear Inverted Pendulum Mode (3D-LIPM), which involves constraining the movement of the CoM to a plane. To implement the ZMP stability criterion with this controller, the distance between the ZMP location provided by the model and a reference ZMP trajectory is used as cost function. Additionally, another term of the cost function will prevent the CoM from moving abruptly.

Once the walking pattern generator is implemented, it will be possible to generate smooth trajectories for the CoM that project the ZMP of the robot over the support polygon. However, this is not sufficient to obtain a robust gait on a real robot. The drawback of the use of the inverted pendulum model is the neglect of the rotational inertia of the parts of the robot. This assumption is acceptable in controlled environments, where external disturbances are limited, but general applications involve the presence of unexpected collisions. Moreover, the approach employed assumes that the soles of the robot are always parallel to the ground and that the actuators perfectly track the reference positions. None of these premises is strictly valid. Therefore, we will use the CoM trajectory provided by the walking pattern generator as a base trajectory that will subsequently be modified by feedback controllers to adapt to disturbances or unexpected states of the system. Following an analysis of the problem caused by the rotational dynamics we will describe two feedback controllers that can counteract the simplifications of the model to generate a robust

gait on a physical robot.

The first controller that we present deals with the rotational dynamics of the robot. When a humanoid robot is disturbed in the upper part of its body, it immediately starts rotating around an edge of its feet. At that instant the robot acquires a certain rotational inertia and the ZMP disappears. Additionally, the size of the support polygon reduces dramatically to the contact between the edge of the sole and the ground and generates a torque that affects the rotational velocity of the body. Since none of these effects are contemplated by the walking pattern generator, they pose a serious threat to the stability of the platform. Thanks to the study of the effect of the CoM acceleration on the torque generated around the edges of the sole, we can design a controller that allows reducing the undesired angular momentum of the platform and notably improving its stability.

There are occasions, however, when the angular momentum controller is not capable to stabilize the robot. For instance, when the motions requested by the locomotion system become unfeasible following a disturbance. In those cases, another controller is necessary to adapt the duration of the steps to the state of the CoM. By means of the 3D-LIPM model, and given the state of the CoM, it is possible to predict the optimal instant of support foot exchange. In other words, to determine the optimum step duration. Nevertheless, in realistic situations, the accurate estimation of the pose of the robot in a fast changing dynamic environment is highly complicated. To achieve a robust reaction to these unexpected system states we use a timing controller that is based on the addition of minimal increments to the duration of the step in every control cycle. In this way, we obtain a robust controller adapted to the modest estimates of the system available. This controller is applied uniquely to the lateral dimension of the CoM trajectory and bases the calculation of the optimum exchange instant in the intersection of the constant energy orbits of the CoM derived from the use of the 3D-LIPM.

Both of the feedback controllers described above require accurate information about the pose of the robot body with respect to the ground, which is assumed to be horizontal. Since the Nao robot is equipped with an inertial unit, it is possible to combine the readings of its gyrometers and accelerometers to obtain an estimation of the orientation of the torso of the robot. Subsequently, by means of the forward kinematics operations, we can obtain the pose of the feet of the robot with respect to the ground frame. In order to

generate the estimation of the orientation of the torso, a sensor fusion approach based on the Kalman filter has been implemented. While gyrometers provide an accurate estimate of the rotational velocity of the torso, the estimate of the angle resulting from the integration of their readings can easily accumulate a significant error. Additionally, the accelerometers can be used to detect the direction of the gravity force, and thus, the direction of the normal to the ground plane. In this case, however, the readings are affected by the collisions between the robot and the environment and the inertia forces, among others. The Kalman filter is therefore configured to fuse measurements from accelerometers and gyrometers and combine the advantages of each sensor. In addition to the readings from accelerometers and gyrometers, the force sensing resistors located at the soles of the robot will be used to detect the contact between the feet and the ground and guide the estimation of the pose of the robot.

Finally we have validated the robustness of the balance control system and studied the stability improvements that it provides both in regular motions and under the action of disturbances. The effect of the angular momentum controller in the balance of the platform is notable even in the absence of disturbances. The reduction of the rotational velocity achieved by the controller involves a drastic enhancement of the stability of the gait, both in the sagittal and in the frontal planes. Besides, the angular momentum controller reveals as a powerful tool to cope with external disturbances. Fundamental as it is the use of the angular momentum controller to improve balance, its action is greatly boosted by the use of a controller that modifies the duration of the steps in anomalous situations. The timing controller that we propose effectively monitors the feasibility of the planned trajectory for the CoM and adjusts the steps duration if necessary. In summary, the experiments demonstrate that the combined use of angular momentum and timing controllers enhances notably the stability of the gait and allows the robot to perform stable motions and to recover from impacts.

Resumen

Para conseguir que los robots humanoides puedan colaborar con los seres humanos y liberarlos de actividades peligrosas o desagradables, hemos de conseguir primero que marchen de forma fiable. En las últimas décadas, el equilibrio de robots humanoides ha atraído la atención creciente de la comunidad científica. Sin embargo, a pesar de que se han logrado importantes resultados en el plano teórico y en simulaciones, el equilibrio de robots humanoides reales continúa siendo objeto de debate. Puesto que estos robots están diseñados para funcionar en ambientes cotidianos, es fundamental que los métodos de control sean validados en plataformas físicas y expuestos a condiciones exigentes. En este trabajo proponemos un sistema de control del equilibrio que mejora la robustez de la marcha y la resistencia frente a perturbaciones. Este método combina el criterio de estabilidad del punto de momento cero con el control del momento angular y de la temporización del paso.

La plataforma robótica empleada para validar estos desarrollos es la versión del año 2010 del robot comercial humanoide Nao, desarrollado por la empresa francesa Aldebaran Robotics. El robot mide 57 *cm* de alto y cuenta con 21 grados de libertad de los cuales 11 están situados en las piernas. El estudio de las características específicas de esta plataforma es un prerrequisito clave para el diseño del sistema de equilibrio. Por lo tanto, comenzaremos revisando pormenorizadamente las especificaciones técnicas del robot humanoide Nao y analizaremos el rendimiento de sus sensores y actuadores. Este análisis permitirá determinar el retardo sensorimotor y el nivel de holgura en el torso. Además, veremos que no es posible el uso de los sensores de fuerza de la planta del pie para estimar la posición del centro de presión. Para completar el análisis de la plataforma, revisaremos las operaciones necesarias para resolver el problema cinemático directo y propondremos una solución para el problema cinemático inverso de esta plataforma. La innovadora configuración de las piernas del robot Nao, con los actuadores yaw-pitch de la cadera acoplados, requiere el uso de un método de

resolución computacionalmente intenso o el desarrollo de una solución específica al problema de la cinemática inversa. Optaremos por esta última opción y propondremos una solución consistente en tres fases. Las dos primeras fases utilizarán controladores tipo p para adaptar las entradas a los requerimientos de la tercera etapa. Esta última etapa será resuelta de manera analítica. El método resultante es tanto computacionalmente eficiente como preciso.

Posteriormente describiremos los módulos requeridos para generar el patrón de marcha, es decir, la secuencia de configuraciones del robot que permitirá alcanzar una cierta posición o velocidad objetivo. Mediante el uso del módulo de cinemática inversa mencionado anteriormente, es posible especificar estas configuraciones de manera conveniente en el espacio cartesiano. Las configuraciones del robot quedan, por tanto, definidas por la configuración de los pies, la localización del centro de masas (*Center of Mass, CoM*) y la orientación del torso, que mantendremos vertical.

La primera fase para generar el patrón de marcha consiste en obtener la posición de las pisadas a partir del tipo de movimiento requerido. Encontrar la localización óptima de una pisada es un problema complejo que aún no ha sido resuelto. De este modo, numerosos desarrollos utilizan métodos empíricos para obtener la posición de las pisadas. En este trabajo, utilizamos una combinación de criterios heurísticos y analíticos para resolver este problema. El tamaño óptimo del paso depende de varios factores. Uno de estos factores está relacionado con la inercia lineal y rotacional del robot: en caso de que ocurran cambios abruptos en la dirección del movimiento, los mecanismos de control del equilibrio pueden llegar al límite de su capacidad para estabilizar el robot. Otro factor que ha de ser considerado para la planificación de las pisadas es el alcance de la pierna, ya que el tamaño máximo de un paso depende de la configuración de la cadena cinemática del robot. Tendremos en cuenta, por tanto, estas dos limitaciones para generar la secuencia de pasos futuros.

Obtenida la posición objetivo para la pisada, nos centraremos en la trayectoria de la zancada que llevará el pie de avance hasta su destino. Las trayectorias del pie de avance pueden inducir un momento de fuerza sobre el cuerpo del robot debido a las fuerzas de reacción generadas por las colisiones contra el suelo y a las fuerzas de inercia que aparecen cuando el pie es acelerado. Además, estas fuerza de inercia pueden impedir que los servomotores sigan adecuadamente las

trayectorias de referencia de las articulaciones. En este trabajo, las trayectorias del pie de avance son ejecutadas en bucle abierto y las suelas de los pies se mantienen paralelas entre sí. La implementación de la trayectoria hexadimensional del pie de avance es determinada mediante curvas cúbicas de tipo Bézier definidas por partes. El diseño de estas trayectorias tiene por objetivo evitar las colisiones contra el suelo que se producen típicamente en el talón o en la punta del pie. Este sistema permite la modificación de la posición de la pisada en tiempo de ejecución y proporciona flexibilidad para generar trayectorias complejas.

El movimiento del CoM también contribuye de forma determinante al equilibrio de los robots bípedos. Aunque el diseño de las trayectorias del CoM continúa siendo activamente investigado, podemos identificar el uso del punto de momento cero (*Zero Moment Point, ZMP*) aplicado a un modelo de péndulo invertido como elementos comunes en la mayoría de los sistemas de locomoción. Este criterio establece que la estabilidad del robot en cada instante está condicionada por la existencia del ZMP dentro del polígono de soporte del robot. Para generar patrones de marcha estables para el robot en un instante determinado es necesario, por tanto, generar trayectorias para el centro de masas que permitan proyectar el ZMP en el polígono de soporte del robot en ese instante y en el futuro. Para resolver este problema contamos con un tipo de controlador que proporciona una solución óptima para un intervalo futuro limitado: el control predictivo basado en modelo. Este controlador confía en un modelo del sistema dinámico y minimiza una función de coste a lo largo de un horizonte deslizante. De este modo, el controlador predictivo consigue optimizar la acción de control en un instante, teniendo en cuenta la evolución futura del sistema. Puesto que el control predictivo ofrece sus mejores resultados cuando se aplica a modelos lineales, el modelo del péndulo invertido no puede ser utilizado directamente. En su lugar, empleamos el péndulo invertido tridimensional en modo lineal (*Three Dimensional Linear Inverted Pendulum Mode, 3D-LIPM*), que limita el movimiento del CoM a un plano. Para implementar el criterio de estabilidad del ZMP en este controlador, utilizamos como función de coste la distancia entre la posición del ZMP proporcionada por el modelo y la trayectoria de referencia del ZMP requerida. Adicionalmente, otro término de la función de coste minimiza la derivada de la aceleración del centro de masas para suavizar su trayectoria.

Una vez implementado el generador de patrones de marcha, será posible obtener suaves trayectorias del centro de masas que proyecten el ZMP sobre el polígono de soporte. Sin embargo, esto no es suficiente para obtener un sistema de locomoción robusto en una plataforma real. Un inconveniente del uso del modelo del péndulo invertido es la ausencia de representación de la inercia rotacional del cuerpo del robot. Esta simplificación es aceptable en entornos controlados donde las perturbaciones externas sean muy limitadas, pero las aplicaciones de uso general implican la presencia de colisiones inesperadas. El sistema de generación de patrones de marcha basado en el 3D-LIPM que empleamos asume que las suelas de ambos pies son paralelas entre sí y se apoya firmemente en el suelo, y que los servomotores son capaces de seguir de forma precisa las posiciones de referencia. Ninguna de estas premisas se cumple de manera estricta. Además, el generador de patrones de marcha no tiene en cuenta las perturbaciones e ignora la inercia de las partes del robot. Por lo tanto, utilizaremos la trayectoria del centro de masas proporcionada por el generador de patrones de marcha únicamente como trayectoria base, que será posteriormente modificada por controladores retroalimentados para adaptarla a las perturbaciones o a los estado inesperados del sistema. Tras analizar el problema causado por las dinámicas rotacionales, describiremos dos controladores retroalimentados capaces de contrarrestar los efectos perjudiciales de las simplificaciones del modelo para generar una marcha robusta en un robot real.

El primer controlador que presentamos se encarga de las dinámicas rotacionales del robot. Cuando un robot humanoide recibe una perturbación en la parte superior de su cuerpo, éste rota alrededor de uno de los bordes de sus suelas. Inmediatamente tras la perturbación, el robot adquiere una cierta cantidad de inercia rotacional y el ZMP desaparece. Asimismo, el tamaño del polígono de soporte se reduce drásticamente al área de contacto entre el suelo y el borde la suela. En este punto aparece un momento que afecta a la velocidad angular del robot. Puesto que ninguno de estos efectos están contemplados por el generador de patrones de marcha, suponen un seria amenaza para la estabilidad de la plataforma. Gracias al estudio del efecto de la aceleración del centro de masas sobre el momento generado alrededor del borde de la suela, podremos diseñar un controlador que permitirá reducir el indeseable momento angular de la plataforma y mejorar notablemente su estabilidad.

Hay ocasiones, sin embargo, en las que el controlador de momento angular no es capaz de estabilizar el robot. Por ejemplo, si los movimientos requeridos por el sistema de locomoción han dejado de ser factibles. Para estos casos, es necesario utilizar otro controlador que adapte la duración de los pasos al estado del sistema, representado por la posición, velocidad y aceleración del centro de masas. Utilizando el modelo 3D-LIPM y partiendo del estado estimado del centro de masas, es posible predecir el instante óptimo en el que el soporte del robot debería ser transferido de un pie al otro. O dicho de otro modo, la duración del paso. Sin embargo, en circunstancias realistas, la estimación precisa de la posición y orientación del robot respecto al suelo es complicada. Para conseguir reaccionar de manera robusta a estados inesperados del sistema utilizaremos un controlador de la temporización basado en la adición de incrementos mínimos a la duración del paso en cada ciclo de control. De este modo conseguiremos un controlador robusto adaptado a las precarias estimaciones del estado del sistema disponibles. Este controlador se aplica únicamente al plano frontal y basa el cálculo del instante óptimo de transición en la intersección de las trayectorias de energía orbital constante que se derivan del uso del 3D-LIPM.

Los dos controladores retroalimentado descritos en este trabajo requieren información precisa acerca de la orientación del cuerpo del robot con respecto al suelo, que se asume que es horizontal. Puesto que el robot Nao está equipado con una central inercial, es posible combinar las medidas de los acelerómetros y girómetros para obtener una estimación de la orientación del torso. Posteriormente, mediante las operaciones de cinemática directa podemos deducir la posición de los pies con respecto al plano del suelo. Para obtener la estimación de la orientación del torso, implementaremos un método de fusión sensorial basado en el filtro de Kalman. Mientras que los girómetros proporcionan una medida precisa de la velocidad angular del torso, la estimación del la orientación del torso que resulta de integrar sus lecturas puede acumular fácilmente un error significativo. Por otro lado, los acelerómetros permiten detectar la dirección de la fuerza de la gravedad y, por lo tanto, la orientación del plano del suelo. En este caso, sin embargo, las medidas presentan un nivel de ruido significativo, debido a las colisiones entre el robot y el medio y las fuerzas de inercia, entre otras causas. Mediante un filtro de Kalman podremos fusionar las medidas de acelerómetros y girómetros y combinar las ventajas de cada sensor. Además de las lecturas de

los sensores de la unidad inercial, utilizaremos los sensores de fuerza situados en las suelas del robot para detectar los contactos entre los pies y el suelo y guiar la estimación de la orientación del torso.

Por último, validaremos la robustez del sistema de control y analizaremos las mejoras en la estabilidad conseguidas tanto en movimientos estándares como ante perturbaciones. El efecto del controlador de momento angular en el equilibrio de la plataforma es notable incluso en ausencia de perturbaciones. La reducción de la velocidad angular lograda por el controlador implica una mejora drástica de la estabilidad de la marcha tanto en el plano sagital como frontal. Además, el controlador de momento angular se revela como una herramienta poderosa para lidiar con las perturbaciones externas. Aún siendo fundamental el uso del controlador angular para mejorar el equilibrio, su acción es insuficiente si no cuenta con otro controlador que modifique la duración de los pasos en circunstancias anómalas. El controlador de la temporización de los pasos que proponemos monitoriza eficazmente la viabilidad de las trayectorias del centro de masas planificadas y ajusta la duración de los pasos en caso de necesidad. En resumen, los experimentos demuestran que el uso combinado de controladores de momento angular y de temporización de los pasos mejoran notablemente la estabilidad de la marcha y permiten al robot ejecutar movimientos estables y recuperarse de los impactos.

Universidad de Murcia

Facultad de Informática

Robust Feedback Control of Omnidirectional Biped Gait on the Nao Humanoid Robot

Control Retroalimentado Robusto de Marcha Bípeda Omnidireccional en el Robot Humanoide Nao

Tesis doctoral

Realizada por:

Juan José Alcaraz Jiménez

Supervisada por:

Dr. Humberto Martínez Barberá

Dr. David Herrero Pérez

2014

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In order to allow humanoid robots to participate in a broad range of everyday activities, it is necessary to develop smooth and reliable gaits. In last decades, the balance of humanoid robots has attracted an increasing attention of the scientific community. However, even though important results have been achieved in the theoretical level and in simulations, the balance control in physical robots still remains an open issue. In this work we present a balance control approach for the Nao robot that takes into account the special features of this platform and that allows robust walking and disturbance rejection.

1.1 Objectives and methodology

Back in 2008, the French company Aldebaran Robotics released the first commercial version of its humanoid robot Nao. The demands to acquire some units of this new affordable humanoid were quite elevated. However, the University of Murcia gained the right to access to two units of the first batch produced, thanks to its research track record in the field of service robotics.

The range of activities that were intended for the new robots, which included soccer competitions, required a dexterous and robust locomotion system. Even though the manufacturer of the robot already delivered some basic locomotion routines in these primitive versions, the software modules were not reliable. Subsequent versions dramatically improved the performance of the locomotion system but still did not match the achievements of independent developments, that were progressing fast. There was undoubtedly room for improvement. Thus was born this work, from the need to provide the Nao platform with a dexterous, robust and omnidirectional gait.

As a first attempt, we implemented several versions of open-loop walk engines obtained from the existing literature. However, the ample backlash of the

platform hindered the accurate reproduction of movements required to ensure stability according to kinematic models. For this reason, we decided to focus our work on the development of a robust balance control approach.

In this way, we selected a model-based walking engine for the generation of the omnidirectional walking pattern that would provide the base trajectories for the gait. Once again, the challenge was in finding control strategies that could cope with the limited performance of the sensors available in the Nao platform. Most of the existing approaches required sophisticated sensors to accurately estimate the pose of the robot or the location of the center of pressure. At this point, it became evident that we would need new control strategies to adapt our walking pattern generator to the special features of the robot. Moreover, an important advantage of developing a balance control approach based on reduced sensorial information lies in its compatibility with a broad range of platforms. Summarizing, the objectives for this thesis are:

- Compile a set of approaches that allow omnidirectional robust gait for the humanoid Nao.
- Design a set of balance controllers adapted to the performance of the Nao platform.
- Experimental validation of the locomotion system obtained.

Once the objectives are set, we will begin this work with an evaluation of the performance of the sensors and the actuators of the robot. The conclusion drawn in this examination will allow us to configure appropriately the walking pattern generator, which will be selected to provide omnidirectional gait. Next, the rotational dynamics of the system will be examined to design a controller that manipulates the CoM to reduce the rotational velocity. On the other hand, the orbital energy concept will be employed to calculate the appropriate range of time for the step duration and adapt it online. Finally, experiments will be carried out in the physical robot to validate that the approach developed improves the stability of the gait and the disturbance rejection capabilities.

1.2 Main contributions

In this thesis we describe a complete approach to design the gait for a position controlled humanoid robot. The development of a gait for a humanoid robot is a complex task that needs to be decomposed in simpler stages. This work offers a reasoned selection of the appropriate approach selected for every module to achieve a robust locomotion system. Although this compilation of approaches is

of interest in itself, the most important contribution of this work concerns the specific areas were innovative solutions are presented. In this manner, the main contributions are:

- Angular momentum controller to reduce the rotational velocity of the body of the robot around the contacts points between the feet and the ground. The controller is inferred from the study of the rotational dynamics of the robot and is based on the effect of the acceleration of the CoM on the rotational velocity.

- Timing controller to adapt footstep durations in real time. This controller monitors the dynamics of the CoM and the target trajectories for the zero moment point. The controller only actuates if the reference trajectories for the zero moment point cannot be tracked accurately and there is a risk for the stability of the gait. To estimate the timing requirements of the gait, reference orbital energy levels are employed.

- Inverse kinematics solution for Nao. The innovative configuration of the legs of Nao, with two coupled yaw-pitch joints, requires the use of a new inverse kinematics approach. Our solution consists of three stages. The first two stages use p-controllers to adapt the inputs to the third stage, which is solved analytically. The resulting approach is both computationally efficient and accurate.

- Sensor fusion approach to estimate the orientation of the body. The Kalman filter employed fuses measurements of accelerometers and gyromenters of the inertial unit and the contact information provided by the FSRs to estimate the orientation of the torso. Subsequently, through the use forward kinematics operations, we obtain the pose of the feet of the robot in the ground frame.

1.3 Outline of this thesis

Below we provide a brief overview of the content of each of the chapters of this thesis:

In **Chapter 2** we review the relevant literature for this thesis. After contextualizing the problem to solve, we describe several concepts and stability criteria related to biped walking. Then, we present the most important humanoid models and walking pattern generators.

In **Chapter 3** we analyze the features of the Nao robot. We first provide a general picture of the technical specifications of the platform. Then, we carry

out a detailed analysis of the performance of the sensors and actuators of the platform. Finally, we describe the procedure to solve the forward kinematics problem and a new approach to solve the inverse kinematics problem.

In **Chapter 4** we describe the combination of approaches that will be used to generate the target trajectories for the CoM and the swing foot of the robot. Additionally, we discuss the performance of the resulting walking pattern generator to create omnidirectional gaits.

In **Chapter 5** we present two novel feedback controllers for the balance of biped robots and an aproach based on the Kalman filter to fuse sensorial information from several sources. Prior to the design of the angular momentum controller, we analyze the rotational dynamics of a robot modeled as an inverted pendulum and the problems derived from neglecting it. Then, we obtain the equations for the angular momentum controller. At last, we exploit the concept of orbital energy to derive a step timing controller.

In **Chapter 6** we validate the performance of the proposed balance controllers in a range of situations. We measure the impact of the controllers in the stability of the gait as well as their performance against disturbances.

Chapter 7 concludes this thesis. We provide a brief summary of the results and outline the potential future works.

Chapter 2

Balance Control of Humanoid Robots: State-of-the-Art.

2.1 Introduction

In last decades, the vertiginous pace of technological innovations is changing radically our way of life. Advances in computing hardware, software and telecommunications allowed the access to powerful services and abundant information to anybody from their home computers. Recently, the apparition of smart-phones has permitted to extend this access and make it available from almost any place.

However, the capability of mobile devices to alter the physical environment where they are immersed is still limited. The next technological revolution will be certainly concerned with intelligent devices that are able to interact physically with the environment. That is, with service robots.

Humanoid robots are a type of service robots that mimic the appearance of the human body. In this manner, these robots employ hand-like actuators to manipulate the objects in their surrounding and are equipped with two legs that use to propel themselves. Both activities involve a fabulous scientific challenge, because of the integration of technologies that require. For instance, performing grasping and walking requires the command of simpler tasks like perception, planning and control.

In this work we focus on the locomotion activity. Since humanoid robots employ legs to interact with the environment, they depend on intermittent contacts to propel and sustain themselves. This fact makes of biped walking a highly unstable maneuver and hence the robustness of the walking approach is a paramount feature. Indeed, the fall of a robot might involve catastrophic consequences for the integrity of both the robot and its surrounding environment.

The study of balance control is therefore a relevant topic in robotics research. In this chapter we will review the main issues and the literature concerning balance in humanoid robots. To begin with, we will discuss the motivation to develop humanoid robots and describe the most advanced models. Next we will enumerate the different simplified models that are usually employed to recreate the dynamics of humanoid robots. This is followed by a review of the criteria that are used to ensure analytically the balance of the gaits. The next section is devoted to compile the main types of approaches that are employed to generate the walking pattern of the robots. While some of these walking engines employ analytic equations based on the models, others are based on pure empirical tuning. In any case, all walking patterns can benefit from the use of the feedback received from the sensors. The processes to adapt the walking patterns according to the sensory readings is reviewed in the control balance section. Finally, we compile the type of sensors involved in the perception of the dynamics of the body of the robot and describe the process to fuse the redundant information that they provide.

2.2 Humanoid Robots

The word *robot* was first presented to the public in the play R.U.R. (Rossum's Universal Robots) by the Czech writer Karel C̃apek in 1920. This term comes from the Czech word *robota*, meaning "hard work". Originally, *robota* was the period of the year that a serf had to work for his lord. Nevertheless, it was Isaac Asimov who popularized the term in his science fiction novels and short stories from the 1940s decade. Nowadays, the term *robot* has evolved and it is becoming difficult to define. According to the Encyclopaedia Britannica, a robot can be described as follows:

> "*A robot is any automatically operated machine that replaces human effort, though it may not resemble human beings in appearance or perform functions in a human-like manner*".

Humanity has been since long interested in the possibility to create automatic devices. In this way, there are records of automatic machines powered by air pressure, steam and water in ancient Greek and Chinese civilizations. But it is not until the 20th century when the technology grows mature enough to allow the creation of robots capable of uses other than entertaining.

In this way, in 1954 the company Unimate presents the first commercial robot, which is conceived to lift hot pieces of metal and stack them. Since then until present, the presence of robots in factories did not stop increasing, specially in

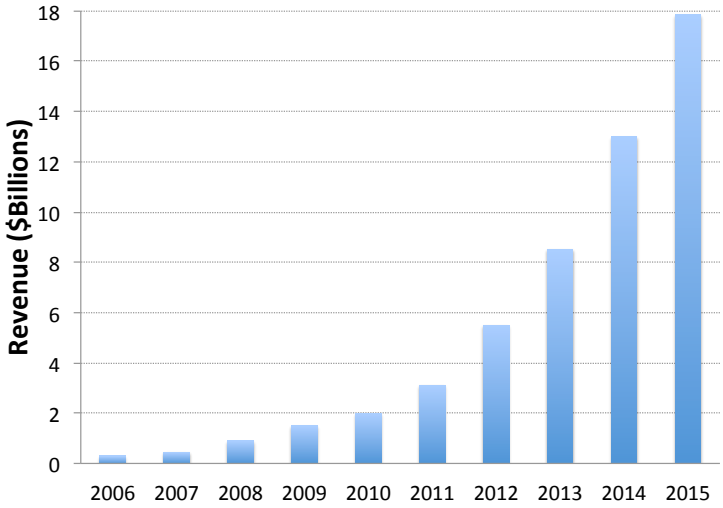


Figure 2.1: Service robotics market outlook. Source: Grishin Robotics [31].

the automotive industry. This type of robots, employed to perform automatic tasks in factories, are known as industrial robots.

Industrial robots have remained for long the main market for robotic applications and have been therefore the fundamental target of research efforts. At present, however, a new type of robots has emerged that attracts increasing attention from the scientific community: the service robot.

According to the International Federation of Robotics [42], service robots can be defined as follows:

> "*A service robot is a robot which operates semi- or fully autonomously to perform services useful to the well-being of humans and equipment, excluding manufacturing operations.*"

Service robots are expected to assist humans in activities such as agriculture, domestic tasks, leisure, medicine, guiding, rescue, surveillance and care of elderly or handicapped people. Since these activities are generally performed in unstructured environments, service robots must face the problem of perceiving the environment and adapting to it, which poses a significant challenge. That is the reason why the development of commercial service robots is delayed in relation to that of industrial robots. However, the prospects of the service robotics market show a favorable trend for the next years, as depicted in Figure 2.1.

Service robots, in their turn, subdivide in several categories according to their shape. One of these categories is humanoid robots, which are those whose body shape is built to resemble that of the human beings. The resemblance between humanoid robots and humans involves certain benefits. First, they provide a

natural interface to interact with humans. Since humans are used to communicate with each other through speech and gestures, a machine that is able to perform these activities in a human-like manner is suitable for an efficient cooperation with humans. In this way, the harmonious look of the robot and the availability of a sufficient number of joints to configure the arms, hands, and face to create realistic expressions is important.

Additionally, the human shape of the body is also beneficial for the interaction with the environment. Over millions of years, humans have created tools that adapt to their body and expand their capabilities. While machines that are conceived to accomplish a certain task in a highly efficient way would benefit from the use of a specific actuator, a robot equipped with a human-like hand would enjoy a much higher versatility. It is difficult to picture an extremity, other than a hand, that would permit the robot to accomplish activities like wielding a pair of scissors, playing the keys of a piano and opening the lid of a jar, just to name a few examples.

The benefits of the adaptation to humans tools and environments is not limited to the upper part of the body. Legs also play an important role in increasing the versatility of robots. Although wheeled machines are able to move in a highly efficient way along even surfaces, their use is severely limited by obstacles. Although an important effort is being currently made to adapt sidewalks and accesses to public buildings to the requirements of wheelchair users, stairs and steps are still common in human environments, and legs provide an effective mean to surpass them. Likewise, the use of pedals in numerous machines also imposes leg-like extremities.

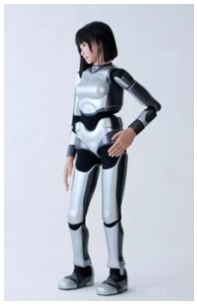
Furthermore, researching on humanoid robots has an additional application whose impact in the quality of life of human beings is extraordinary: the development of medical prosthesis and exoskeletons. As the knowledge about the inner workings of the human body increases and the manufacturing techniques and materials develop, new robotic prosthesis appear that are able to replicate accurately the appearance and performance of human limbs and hence to replace them. In this manner, it is possible to augment both life expectancy and quality of life. Similarly, powered exoskeletons are automatic machines that couple to the body to support its movements. The application of such devices range from the restoration of lost capabilities in disabled patients to the expansion of strength and resistance of workers and soldiers.

However, most of the aforementioned applications of humanoid robots will only become feasible in the mid- or future-term. Actual commercial humanoid robots are usually conceived for education, research or entertainment purposes. In Table 2.2 some of the most popular state-of-the-art robots are introduced, shorted by their number of degree of freedom. That is, by the number of joints whose state can be modified to set the robot in a certain pose.

ASIMO 2011 *Advanced Step in Innovative MObility* [40]

- Height: 130 cm.
- Weight: 48 Kg.
- DoF: 57
- Released: 2011
- Manufacturer: Honda Motor Co.

HRP-4C *Humanoid Robotics Project* [3]



- Height: 158 cm.
- Weight: 43 Kg.
- DoF: 42
- Released: 2009
- Manufacturer: AIST and Kawada Industries.

REEM-B [81]

- Height: 147 cm.
- Weight: 60 Kg.
- DoF: 41
- Released: 2008
- Manufacturer: Pal Robotics.

QRIO *Quest for cuRIOsity* [91]

- Height: 60 cm.
- Weight: 7.3 Kg.
- DoF: 38
- Released: 2005
- Manufacturer: Sony Corporation.
- Development was halted in 2006

HOAP-3 *Huamnoid for Open Architecture Platform* [20]

- Height: 60 cm.
- Weight: 9 Kg.
- DoF: 28
- Released: 2005
- Manufacturer: Fujitsu Automation.

NAO Robocup Edition [26]

- Height: 57 cm.
- Weight: 4.3 Kg.
- DoF: 21
- Released: 2008
- Manufacturer: Aldebaran Robotics.

RH-1 [4]

- Height: 152 cm.
- Weight: 50 Kg
- DoF: 21
- Released: 2006
- Manufacturer: University Carlos III.

DARwIn-OP *Dyn. Anthropomorphic Robot with Intelligence-Open Platform* [33]

- Height: 45.5 cm.
- Weight: 2.9 Kg.
- DoF: 20
- Released: 2010
- Manufacturer: ROBOTIS, Virginia Tech, Universities of Purdue and Pennsylvania.

Table 2.1: Some state-of-the-art humanoid robots.

2.3 Humanoid Robot Modeling

Humanoid robots have complex bodies with irregular shape and mass distribution. To solve the problem of balance from an analytic point of view, it is advantageous to obtain an elemental representation of the dynamics of the robot described by sets of simple equations. Ideal features of a model are simplicity, both conceptual and mathematical, and accurate representation of the dynamics of the real system. Next we introduce the main approaches employed to model the kinematics of the body of humanoid robots.

2.3.1 Three Dimensional Inverted Pendulum

The Three Dimensional Inverted Pendulum (3D-IP) model involves a great simplification compared to the real body of the robot. Before explaining the model it is necessary to introduce the concept of Center of Mass (CoM). The CoM is the weighted average location of the mass of an object.

Assuming the body of the robot can be modeled as a set of particles, the location of the CoM can be calculated with (2.1), where M is the total mass of the robot, m_i is the mass of the i-th particle and c_i the location of the same particle.

$$C = \frac{1}{M} \sum m_i c_i \tag{2.1}$$

The inverted pendulum model represents the whole body of the robot as a point particle located at the CoM. This particle is linked to the base of the robot by a telescopic massless leg. The position $\boldsymbol{R} = (x, y, z)$ of the CoM can be expressed in terms of its coordinates in the system depicted in figure 2.2:

$$\begin{aligned} x &= rS_p, \\ y &= -rS_r, \\ z &= rD, \end{aligned} \tag{2.2}$$

where $S_r \equiv \sin\theta_r$, $S_p \equiv \sin\theta_p$, $D \equiv \sqrt{1 - {S_r}^2 - {S_p}^2}$.

By naming (τ_r, τ_p, f) the actuator torque and force corresponding to the state variables (θ_r, θ_p, r), the motion equations of the 3D inverted pendulum in

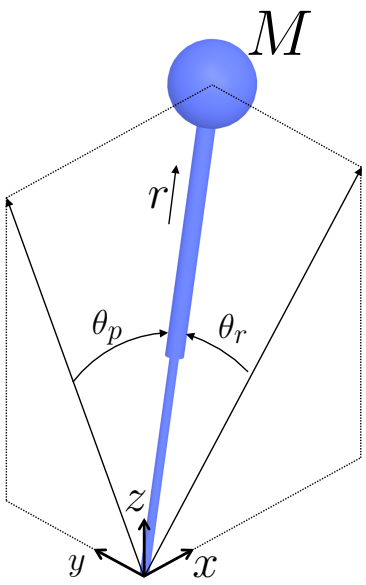


Figure 2.2: The 3D-IP.

Cartesian coordinates become:

$$m\begin{pmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{pmatrix} = (J^T)^{-1}m\begin{pmatrix}\tau_r\\ \tau_p\\ f\end{pmatrix} + m\begin{pmatrix}0\\ 0\\ -mg\end{pmatrix}, \tag{2.3}$$

where m is the mass of the pendulum, g the gravitational acceleration and $\boldsymbol{J}$ is:

$$\boldsymbol{J} = \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{q}} = \begin{pmatrix}0 & rC_p & S_p\\ -rC_r & 0 & -S_r\\ -rC_rS_r/D & -rC_pS_p/D & D\end{pmatrix} \tag{2.4}$$

with $C_r \equiv \cos\theta_r, C_p \equiv \cos\theta_p$. The inverses Jacobian can be removed by multiplying $\boldsymbol{J}^T$ from the left in equation (2.3):

$$m\begin{pmatrix}0 & -rCr & -rC_rS_r/D\\ rC_p & 0 & -rC_pS_p/D\\ S_p & -S_r & D\end{pmatrix}\begin{pmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{pmatrix} = \begin{pmatrix}\tau_r\\ \tau_p\\ f\end{pmatrix} - mg\begin{pmatrix}-rC_rS_r\\ -rC_pS_p/D\\ D\end{pmatrix} \tag{2.5}$$

If the first row of the equation is multiplied by D/C_r we get:

$$m(-rD\ddot{y}) - rS_r\ddot{z}) = \frac{D}{C_r}\tau_r + rS_rmg. \tag{2.6}$$

By translating this equation to Cartesian coordinates it becomes:

$$m(-z\ddot{y} + y\ddot{z})) = \frac{D}{C_r}\tau_r - mgy \tag{2.7}$$

And the result in the sagittal dimension is similar:

$$m(z\ddot{x} + x\ddot{z})) = \frac{D}{C_p}\tau_p - mgx. \tag{2.8}$$

In this way an elemental representation of the dynamics of the body is achieved. Nevertheless, the motion equations in sagittal and lateral planes are interrelated, which complicates the mathematical operations required to apply the model to walking pattern generation.

2.3.2 Three Dimensional Linear Inverted Pendulum Mode

Restraining the movements of the CoM to a plane allows to simplify the motion equations of the inverted pendulum model. Since the CoM height is kept roughly constant during human walking, it is a reasonable assumption to keep it in an plane parallel to the ground, as displayed in Figure 2.3. The resulting model is known as the Three Dimensional Linear Inverted Pendulum Mode (3D-LIPM) [51].

To obtain the simpler set of equations, the first step is to set the restriction of the movement of the CoM to the plane mentioned above:

$$z = k_x x + k_y y + z_c \tag{2.9}$$

Substituting z in (2.7) and (2.8) yields:

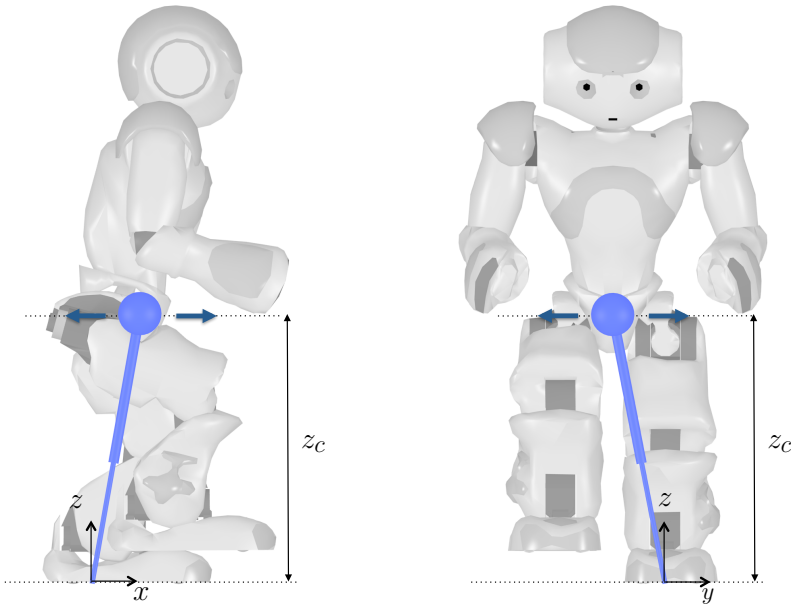
$$\ddot{y} = \frac{g}{z_c}y - \frac{k_1}{z_c}(x\ddot{y} - \ddot{x}y) - \frac{1}{mz_c}u_r, \tag{2.10}$$

$$\ddot{x} = \frac{g}{z_c}x - \frac{k_2}{z_c}(x\ddot{y} - \ddot{x}y) - \frac{1}{mz_c}u_p, \tag{2.11}$$

where u_r and y_p are virtual inputs created to compensate the input nonlinearity.

$$\tau_r = \frac{C_r}{D}u_r \tag{2.12}$$

$$\tau_p = \frac{C_p}{D}u_p \tag{2.13}$$



(a) Sagittal Plane (b) Frontal Plane

Figure 2.3: The 3D-LIPM restricts the movement of the Inverted Pendulum to a plane. Hence motion equations in sagittal and frontal planes can be decoupled.

The plane of movement of the CoM should be parallel to the ground in regular gaits. If the floor is horizontal, the motion equations become:

$$\ddot{y} = \frac{g}{z_c} y - \frac{1}{m z_c} u_r, \tag{2.14}$$

$$\ddot{x} = \frac{g}{z_c} x - \frac{1}{m z_c} u_p. \tag{2.15}$$

On the other hand, if the robot is to climb stairs, the plane of the CoM will have a certain slope ($k_x \neq 0$, $k_y \neq 0$). Multiplying (2.10) and (2.11) by x and y respectively:

$$x\ddot{y} - \ddot{x}y = \frac{-1}{mz}(u_r x + u_p y). \tag{2.16}$$

By introducing an additional constraint:

$$u_r x + u_p y = 0, \tag{2.17}$$

this result provides the same dynamics that equations (2.14) and (2.15). Finally,

if the input torques u_r and u_p are set to zero, the equations that describe the motion of the CoM become:

$$\ddot{y} = \frac{g}{z_c} y \tag{2.18}$$

$$\ddot{x} = \frac{g}{z_c} x, \tag{2.19}$$

This is an interesting result since it permits to describe the motion of the CoM with linear and independent equations.

Moreover, the nature of the movements of the CoM described by these equations is familiar. Since the acceleration of the CoM is proportional to its distance to the pivot point, we could imagine a force acting on the CoM as it approaches the pivot point. In this way, the motion equations of the CoM provided by the 3D-LIPM are similar to the planetary motions described by the Kepler's second law. It is common to both models the existence of a parameter that defines a trajectory of the moving particle. For analogy, this parameter of the 3D-LIPM is named *orbital energy*. The orbital energy of the trajectories of the CoM can be obtained by integrating (2.18) and (2.19).

$$E_y = -\frac{g}{2z_c} y^2 + \frac{1}{2}\dot{y}^2, \tag{2.20}$$

$$E_x = -\frac{g}{2z_c} x^2 + \frac{1}{2}\dot{x}^2 \tag{2.21}$$

2.3.3 Inverted Pendulums with Angular Momentum

The main drawback of the 3D-LIPM model is that it neglects the rotational inertia of the robot, which is a reckless assumption. The inverted pendulum model provides an efficient mean to represent the kinematic behavior of the robot and it is therefore a powerful tool to understand and manipulate the balance of a humanoid robot. However, under some circumstances, it exceeds its simplification task and fails in describing accurately the movement of robots. Indeed, this simplification involves a cost: since the inverted pendulum model assumes that all the mass of the robot is concentrated at the CoM, it neglects the angular momentum of the robot's body around the CoM (Centroidal Angular Momentum). Let us now recall the definition of angular momentum of a system of N particles around a certain axis:

$$\boldsymbol{L} = \sum_{i=1}^{N} \boldsymbol{r}_i \times m_i, \boldsymbol{v}_i \tag{2.22}$$

where $\boldsymbol{L}$ is the total angular momentum, $\boldsymbol{r}_i$ is the distance from the origin to the location of particle i, and m_i and $\boldsymbol{v}_i$ are the mass and velocity of the particle, respectively.

In this way, as the particles move away from the rotation axis or increase their speed, their contribution to the angular momentum is more perceptible. Specially if the mass of the particles is elevated. This is the case, for example, of the swing foot of a walking robot that moves away from the CoM at a considerable velocity. The existence of a centroidal angular momentum involves a tendency of the body to rotate around the CoM. Since this tendency is an important feature of the kinematic behavior of the robot, several models have been suggested to capture its contribution to the dynamics of the robot at a reduced computing cost.

2.3.3.1 GCIPM

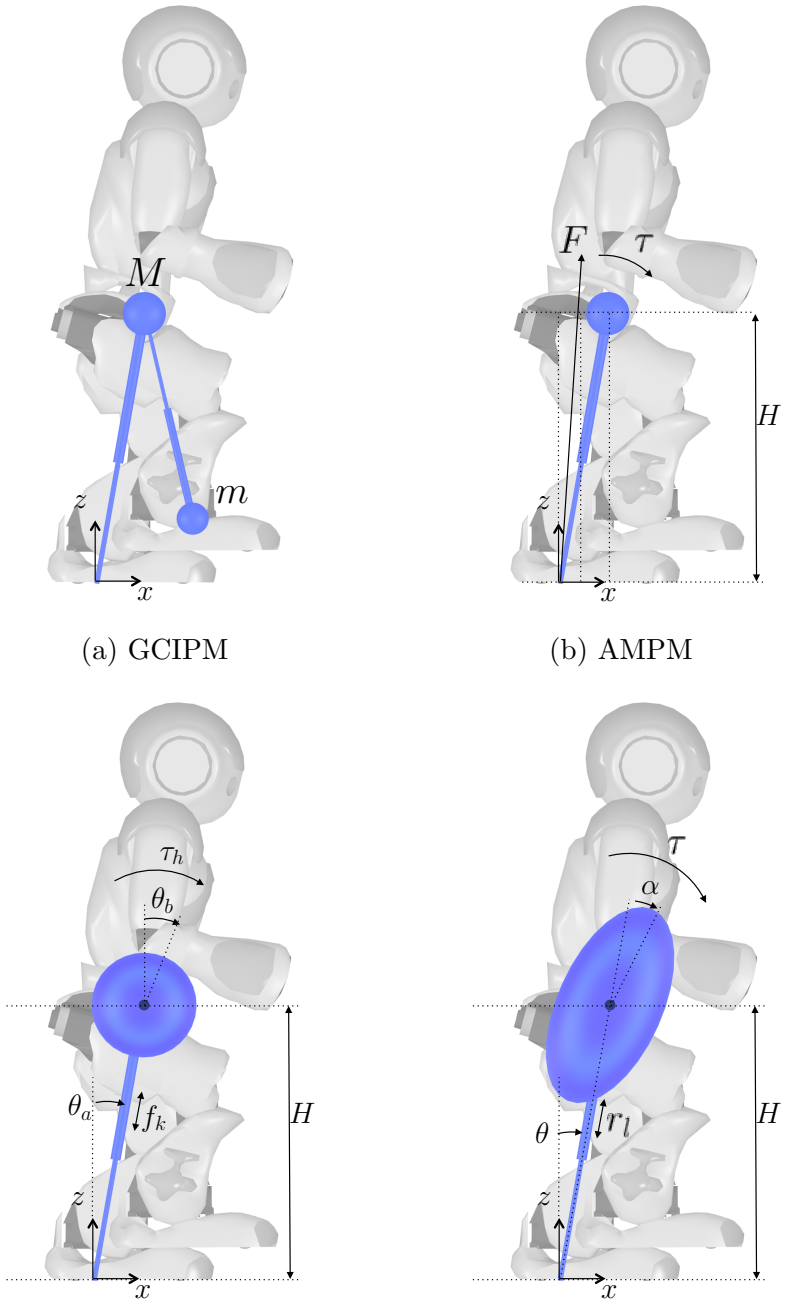
One of these models is the Gravity Compensated Linear Inverted Pendulum Mode (GCIPM) [82, 83]. In this model (Figure 2.4a), a small mass corresponding to the swing leg is added to a linear inverted pendulum model. Since the swing foot moves at a significant velocity and it is at one of the locations that are more distant to the CoM, its contribution to the Centroidal Angular Momentum is important. Therefore, this model provides a more accurate location for the ZMP than the simpler 3D-LIPM, resulting in a better control of the balance of the robot.

2.3.3.2 AMPM

From a different perspective, the Angular Momentum inducing inverted Pendulum Mode (AMPM) [57, 55, 56] also permits to improve the control of the balance of the robot. In this case, the mass model is the same as in the 3D-LIPM, that is, all the mass of the robot is concentrated at a single point located at the CoM of the robot. However, unlike that model, the AMPM allows the reaction force not to pass through the CoM, as depicted in Figure 2.4b. Hence the distance between the reaction force and the CoM generates a Centroidal Angular Momentum that is to be used to correct angular momentum deviations caused by external perturbations.

2.3.3.3 Inverted Pendulum Plus Flywheel

The Linear Inverted Pendulum Plus Flywheel model, [86], is another mean to overcome the problem of the absence of angular momentum representation in the 3D-LIPM. This model incorporates a flywheel to the 3D-LIPM. In this way, the rotational inertia of the parts of the robot's body are included in the model. This



(a) GCIPM (b) AMPM

(c) Flywheel (d) RMP

Figure 2.4: A number of models extend the Inverted Pendulum concept to consider the effect of angular momentum in humanoids.

model is a modified version of the Reaction Wheel Pendulum or the Inertia Wheel Pendulums [79, 92, 9].

However, in [86], the model is employed to calculate the proper location of a capture step that prevents the robot from falling when strongly perturbed. The model, depicted in Figure 2.4c, has two actuators, one to control the motor torque τ_h of the flywheel rotation and another to control the force f_k that modifies the size of the leg. The motion equations are given by:

$$m\ddot{x} = f_x \sin\theta_a - \frac{\tau_h}{l}\cos\theta_a \tag{2.23}$$

$$m\ddot{z} = -mg + f_k \cos\theta_a + \frac{\tau_h}{l}\sin\theta_a \tag{2.24}$$

$$J\ddot{\theta_b} = \tau_h \tag{2.25}$$

where m is the mass of the robot, J the rotational inertia of the flywheel, l the size of the leg, and θ_a is the angle between the vertical axis and the leg and θ_b the angle rotated by the flywheel. In the same way as we did with the 3D-LIPM we will constrain the movements of the CoM to a horizontal plane by setting $\dot{z} = 0$ and $z = z_0$. Then, f_k can be written as

$$f_k = \frac{mg}{\cos\theta_a} - \frac{1}{l}\frac{\sin\theta_a}{\cos\theta_a}\tau_h \tag{2.26}$$

Replacing $\cos\theta_a = z/l, \sin\theta_a = x/l$, we obtain

$$f_k = \frac{mg}{\cos\ theta_a} - \frac{1}{l}\frac{\sin\theta_a}{\cos\theta_a}\tau_h \tag{2.27}$$

Finally, using (2.27) to replace f_k we obtain the equations of motion of the Linear Inverted Pendulum Plus Flywheel Model:

$$\ddot{x} = \frac{g}{z_0}x - \frac{1}{mz_0}\tau_h \tag{2.28}$$

$$\ddot{\theta_b} = \frac{1}{J}\tau_h \tag{2.29}$$

2.3.3.4 Reaction Mass Pendulum

Another step to increase both the complexity of the model and the accuracy in the representation of the dynamics of the system is given with the Reaction Mass Pendulum (RMP), [63]. Because the shape of the body of a humanoid is elongated, the performance of the Inverted Pendulum Plus Flywheel model can be improved by letting the shape of the disc adapt to that of the robot. In this

manner the RMP, Figure 2.4d, comprises a telescopic massless leg joined to an ellipsoid whose size, shape and attitude varies as the robot moves.

2.4 Stability Criteria

Additionally to the models described above, walking pattern generation requires the kinematic conditions to ensure balance. As in the case of body models, balance criteria can be described in different degrees of complexity, depending on the forces and moments that are taken into account. These forces can include gravity, linear inertia, rotational inertia, centrifugal, etc. In this section we will review some stability criteria that yield from the analysis of the forces acting on the humanoid models. In a similar way, in [21] a qualitative comparison of the performance of several criteria is performed.

2.4.1 Static Stability

Static stability is the simplest stability criterion. It indicates whether a still robot is balanced or it will tip over. But before explaining the conditions for stability, the concept of support polygon must be introduced: the support polygon is the minimal convex set (convex hull) that includes the contact area between the robot and the ground

Hence a biped pose is said to be statically stable [90] if the ground projection of its center of mass falls within the support polygon. This criterion can be extended to moving robots, but in that case forces other than gravity will appear and be neglected by this criterion. The more pronounced these forces are, the higher is the error in the balance certainty provided by this criterion.

2.4.2 Zero Moment Point

Nevertheless, robots are usually required to move and hence a more complex criterion is required. As Pai et al. demonstrated in [80], it is not necessary to maintain the CoM above the support polygon to achieve balanced movement. For that reason, a new stability criterion had to be defined to take into consideration the forces that appear when the robot is moving and to ensure dynamic balance. The Zero Moment Point (ZMP) criterion has been successfully employed in a broad range of projects for this purpose.

The fundamental idea of this criterion is that in order to preserve the dynamic balance of a robot during the gait, the whole foot area and not only an edge must be in contact with the ground. In this way, the first step to find the ZMP is to replace all the forces that push the foot towards the ground by a total force

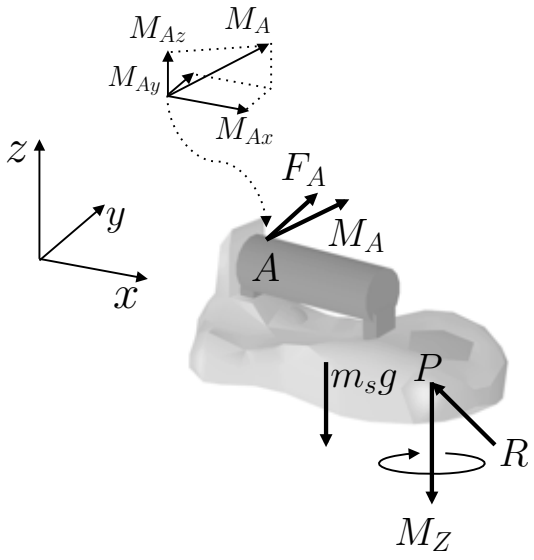


Figure 2.5: Forces and moments acting on the support foot.

acting at the Center of Pressure (CoP). Then, if this total force matches all forces acting on the support foot (gravity, inertia, Coriolis and centrifugal forces and moments), the ZMP exists and is located at the CoP [103]. If the gait is not balanced, on the other hand, the ZMP does not exist, and the robot will tip over around the edges of the support polygon.

The ZMP can be interpreted in another way, depicted in Figure 2.5. Let P be the point that makes $T_x = 0$ and $T_y = 0$. T_x, T_y represent the moments around x- and y-axis generated by the reaction force F_r and the reaction torque T_r, respectively. The point P is the ZMP and can only exist within the domain of the support surface, S:

$$P = (x_{ZMP}, y_{ZMP}, 0) \in S. \tag{2.30}$$

In that situation, the contact between the ground and the support leg is stable, no rotation around the edges of the support polygon occurs.

It is important to notice that we are making a distinction between the ZMP and the CoP. While the location of the ZMP is always the same than the location of the CoP (dynamically balanced gait), the CoP does not always match ZMP (dynamically unbalanced gait). This distinction has been a controversial matter in last decade. While Ambarish claimed in his work [23] of 1999 that ZMP and CoP where equivalent concepts, Vukobratovic defended in 2004 [102] that they should not be confused. In this work we stick to Vukobratovic's point of view.

Even though the ZMP has proven its success and enjoyed a formidable popularity, a number of authors have pointed to its limitations. In this manner,

Wieber demonstrated in [104] that a robot whose support foot swings can be stable. In fact, this results to be a common situation to stable gaits. It is therefore an exaggerate simplification to classify the situations where the ZMP does not exist in the surface of the support polygon as "unstable".

2.4.3 Foot Rotation Indicator

The main drawback of the ZMP is therefore that it can only exist when the soles lie firmly flat on the ground. In real environments, however, robots suffer from disturbances that tilt their soles and face uneven floors. It is therefore required a more complex description of dynamic balance that can be employed in these situations. The Foot Rotation Indicator (FRI) [23], also know as Fictitious Zero Moment Pointm, is a natural extension of the ZMP that achieves this goal. It is defined as the point on the foot/ground contact surface, within or outside the convex hull of the foot support area, at which the resultant moment of the force/torque impressed on the foot is normal to the surface.

The FRI has some advantages over the ZMP:

- The FRI point indicates the occurrence of foot rotation
- The location of the FRI point indicates the magnitude of the unbalanced moment on the foot.
- The FRI point indicates the direction of foot rotation.
- The FRI point indicates the stability margin against foot rotation of the robot.

In view of that set of advantages, it could be expected that the use of the FRI point would take over that of the ZMP. The truth is, however, that the ZMP is still the most frequently employed concept. This is due to the fact that most of the walking pattern generators assume that the feet lie always flat on the ground and execute emergency movements to restore that situation when an abnormal pose is detected. Because the advantages provided by the FRI are only obvious in situations were the robot rotates around the edges of the sole, the adoption of the FRI in those approaches does not involve significant benefits.

2.4.4 Strong and Weak Stability

In order to apply the ZMP to coordination between arms and legs, the concept is redefined in [34] from a different perspective, which gives rise to the Generalized Zero-Moment Point (GZMP). This concept still evolves in the version proposed

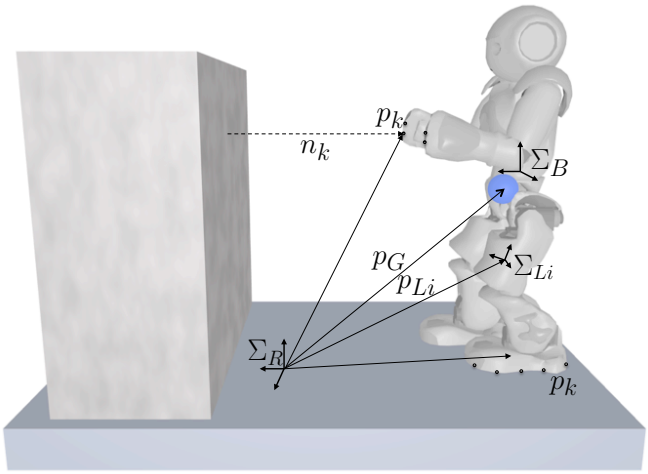


Figure 2.6: Walking robot with multiple contact points with the environment.

in [38], resulting in a very complex and accurate stability criterion. This new criterion considers friction with the sole, unevenness of the floor and contacts of the arms.

Figure 2.6 depicts a sample case where the hands of the robot contact with the environment. The global reference frame is Σ_R, the frame Σ_B is fixed to the torso of the robot and the frames Σ_{Li} are located at the CoM of the i-th links of the robot. Each link has a mass, m_i, an inertia tensor, $\boldsymbol{I}_i$, and an angular velocity, $\boldsymbol{\omega}_i$, referenced in the Σ_R frame. The points $\boldsymbol{p}_k$ define the limits of the support polygons of hands and feet and $\boldsymbol{p}_G$ is the position of the CoM. The force applied at $\boldsymbol{p}_k$ is $\boldsymbol{f}_k$, and $\boldsymbol{n}_k$ is the unit normal vector at $\boldsymbol{p}_k$ pointed to the robot.

The first step to determine stability is to calculate the total force applied by gravity and linear inertia forces at the CoM, $\boldsymbol{f}_G$, and the total moments about the CoM, $\boldsymbol{\tau}_G$, with respect to Σ_R.

$$\boldsymbol{f}_G = \boldsymbol{M}(\boldsymbol{g} - \ddot{\boldsymbol{p}}_G), \tag{2.31}$$

$$\boldsymbol{\tau}_G = \boldsymbol{p}_G \times \boldsymbol{M}(\boldsymbol{g} - \ddot{\boldsymbol{p}}_{\boldsymbol{G}}) - \dot{\boldsymbol{\mathcal{L}}} \tag{2.32}$$

where M is the mass of the robot, $\boldsymbol{g} = (0, 0, -g)$ the gravity vector and $\boldsymbol{\mathcal{L}}$ the angular momentum of the robot with respect to the CoM, defined by:

$$\boldsymbol{\mathcal{L}} = \sum_{i=1}^{N} \{m_i(\boldsymbol{p}_{L_i} - \boldsymbol{p}_G) \times \dot{\boldsymbol{p}}_{L_i} + \boldsymbol{I}_i \boldsymbol{\omega}_i\}. \tag{2.33}$$

On the other hand, the contact force $\boldsymbol{f}_c$ applied from the environment to the robot with respect to Σ_R generates a moment $\boldsymbol{\tau}_C$:

$$\boldsymbol{f}_C = \sum_{k=1}^{K} \sum_{l=1}^{L} \epsilon_k^l (\boldsymbol{n}_k + \mu_k \boldsymbol{t}_k^l) \tag{2.34}$$

$$\boldsymbol{\tau}_C = \sum_{k=1}^{K} \sum_{l=1}^{L} \epsilon_k^l \boldsymbol{p}_k \times (\boldsymbol{n}_k + \mu_k t_k^l) \tag{2.35}$$

where μ_k is the friction coefficient and ϵ_k^l a non-negative scalar. At each $\boldsymbol{p}_k$ the friction cone is modeled as a L-polyhedral cone. Additionally, $\boldsymbol{t}_k^l$ is a unit tangent vector that makes $\boldsymbol{n}_k + \mu_k \boldsymbol{t}_k^l$ the l-th edge of the polyhedral cone. The magnitude of the force of the l-th edge of the approximated friction cone at the k-th contact point is given by ϵ_k^l.

The contact between the robot and the environment is strongly stable when it is guaranteed that the contact is stable to (f_G, τ_G). The contact is weakly stable when it is possible that the contact is stable to (f_G, τ_G).

2.5 Walk Engines

Walking pattern generators or walk engines are the mechanisms employed to calculate the sequences of reference signals for the actuators to generate the motions of the robot. Targets of these approaches include robustness, smoothness, low energy consumption, velocity and versatility of the gait. In this section we review the types of approaches that can be applied to a position controlled humanoid robot.

2.5.1 Model-Free

A common choice to design a walking pattern generator is to employ a physical model of the robot and subsequently apply a stability criterion to obtain the trajectories of the set of actuators that manage the robot. However, it is also possible to generate the walking pattern generator without any physical model. In this section we will review examples of such model-free walking pattern generators that either aim to imitate the strategies of animals or rely on the skills and intuition of motion designers. The main advantage of model-free approaches is that no knowledge about the physical parameters of robot environment is required. On the other hand, the drawback of such approaches is that the walking pattern generators require significant training or empirical tuning by experts before being applied to a platform.

2.5.1.1 Central Pattern Generators

Since humanoid robots are designed to resemble the shape of the humanoid body, a natural solution to the problem of walking pattern generation is to mimic the approach employed by humans. In humans and other animals, the motion of the limbs are controlled by rhythmic patterned signals with sensory feedback [43]. These signals are generated by neural networks located at a Central Pattern Generator (CPG).

In a similar way, it is possible to design rhythmic signals to control the gait of robotic devices [7]. Usually, these signals are applied to some parameters of the extremities of the robot that are subsequently translated to the joint space. The problem with CPG-based approaches is that in order to find a proper combination of shapes and rhythms a vast space must be explored with a trial-error process. Although obviously the most promising way to solve the problem is with automatic learning techniques, at present the CPGs are frequently tuned manually. In any case, both solutions require a significant amount of time.

2.5.1.2 Imitation

Instead of mimicking the neurobiological approaches used by animals to generate the walking patterns, it is also possible to mimic the walking patterns themselves. A problem with this technique is that the robotic platforms are not equipped with as many degrees of freedom as the real animals, and the materials and mass distributions are different. Therefore, if motion data captured from humans are directly replayed in humanoid robots, the resulting gaits are poorly balanced.

For this reason, imitation is usually restricted to animation of avatars. However, it has also been applied successfully to real humanoids in [11]. In that work, a learning approach which is based on imitation is employed to obtain the walking pattern.

2.5.1.3 Handcrafted Trajectories

Although the use of biological data or structures is a reasonable starting point to generate a walking pattern, the most used model-free approach is simply the direct design of the actuator trajectories by human experts. There exists a number of approaches that facilitate the design of these trajectories. However, they usually share a previous stage where the dimensions of the system are reduced. Otherwise, the design of the motions of the robot in the joint-space becomes extremely tedious.

Let us see an example of CoM trajectory generation [28]. The shape, in Cartesian coordinates, of the oscillatory signal of the CoM, t_{com}, is handcrafted

by mixing three types of periodic signals: s, r and l. The three signals are weighed by x_c, y_c and z_c respectively.

$$t_{com} = \frac{x_c \cdot s(\varphi) + y_c \cdot r(\varphi) + z_c \cdot l(\varphi)}{x_c + y_c + z_c} \tag{2.36}$$

$$s(\varphi) = \sin(2\pi\varphi) \tag{2.37}$$

$$r(\varphi) = \sqrt{|\sin(2\pi\varphi)|} \cdot \mathrm{sgn}(\sin(2\pi\varphi)) \tag{2.38}$$

$$l(\varphi) = \begin{cases} 4\varphi & \text{if } p < 0.25 \\ 2 - 4\varphi & \text{if } p \geq 0.25 \wedge p < 0.75 \\ 4\varphi - 4 & \text{if } p \geq 0.75 \end{cases} \tag{2.39}$$

In this way, the CoM can simply be moved smoothly towards the walking direction with:

$$t_{lin} = \begin{cases} 2\varphi & \text{if } \varphi < 0.5 \\ 2\varphi - 1 & \text{if } \varphi \geq 0.5 \end{cases} \tag{2.40}$$

In that approach, the tools of the motion designers are a set of predefined curves that can be combined to obtain the required shape of the oscillatory signal. By the other hand, in [87] the set of tools available to the motion designers consist of a range of virtual components that interact with the system generating virtual forces. Among the components we may find springs, dampers, dashpots, masses, latches, bearings, non-linear potential and dissipative fields, or any other imaginable component.

Since the procedure employed to tune the parameters of the gait requires a significant amount of time in these approaches, it is paramount to configure an agile strategy to run it. The typical solution involves connecting the platform with a computer where the parameters can be modified online. A more advanced solution is proposed in [69], where the gait parameters are adapted according to the corrective signals emitted by a human equipped with a Nintendo Wiimote commercial game controller.

2.5.2 Zero Moment Point

If the physic parameters of the robots are provided by the manufacturers, which is often the case, the use of a model involves important benefits. With an accurate model, the time required to adapt the parameters of the gait drops significantly. Time savings are specially elevated if the space of motions to generate is ample.

In order to generate a walking pattern from a model of the robot, a certain stability criterion is required. The ZMP was the first stability criterion proposed

for dynamic walking and, as mentioned before, it has since then remained the most popular. However, because the concept can only be applied when the feet of the robot lie flat on the floor, it may seem excessively restrictive. On the other hand, when the ZMP is combined with the 3D-LIPM, it is possible to generate the walking pattern of the platform with a set of linear and independent equations. Since this fact has a considerable impact in the computing cost of the algorithms, the combination of ZMP and 3D-LIPM has allowed to generate walking patterns online in a number of platforms. Indeed, its computing efficiency together with conceptual simplicity have been determinant in the level of popularity achieved.

2.5.2.1 Offline-Generated Trajectories

When the first gaits based on the ZMP appeared back in the 80's they did not enjoy a high degree of computational efficiency. Moreover, the computing resources available at that time were also considerably less performing than those of our days. For this reason, trajectories of the actuators had to be computed in advance and subsequently replayed in the robot.

The first biped robot that was able to walk dynamically was developed in 1984 by Takanishi and Kato in the Wasada University of Japan. The approach employed to generate the walking pattern of the robot was based on the transformation of the trajectory of the ZMP into a set of Fourier series [108, 96, 65]. The approach proposed by Nagaska et al. [76], on the other hand, was based on the optimal gradient method. Another example of offline generated trajectories can be found in [89], and even the first versions (year 1997) of the state-of-the-art robot ASIMO required the walking pattern to be generated offline [36]. In following years, however, most of the approaches based on the ZMP would use online generated walking patterns. This was mainly due either to the improvements in the computational resources available at the robots or to the use of more efficient approaches. In this way, in [44] a simplified model with error compensation allowed to generate several motions on the H5 robot, and in [95] the motion of the CoM was generated through the manipulation of the ZMP.

2.5.2.2 Analytical Trajectories with 3D-LIPM

It was not until the introduction of the 3D-LIPM that online generation of walking patterns became available to most platforms. By placing the pivot of the inverted pendulum at the target location of the ZMP, it is possible to obtain the trajectory of the CoM given a certain ZMP reference. Although severely restrictive and inaccurate, the 3D-LIPM permits to generate walking patterns that are acceptably balanced. However, this approach must rely on the presence

of robust systems for balance control.

The results derived from the study of the 3D-LIPM, equations (2.14) and (2.15), show that the motion equations corresponding to sagittal and lateral planes can be separated and that these equations are linear. In this way, once the location of the ZMP is specified, for a given initial state of the CoM there exists a unique solution for its future trajectory. If the pivot point of the pendulum is fixed on the floor during the single step duration, then a variety of different solutions can be made to link one step to the next.

The solution proposed by Harada et al. is to obtain the positions of the footsteps directly from the required speed for the robot [35]. Then, use the 3D-LIPM to generate the trajectories during the single support phase and spline curves to move the CoM from the final state of one step to the initial state of the next one. It is assumed that the double support phase enlarges the support polygon and allows secure movements of the pivot point of the pendulum.

The main advantages of this solution are the possibility to define the exact position of the footsteps and the reduced computing requirements. Moreover, it is possible to obtain smooth transitions of the ZMP from one foot to the other, which avoids the instability produced by strong changes in the acceleration of the CoM. This solution is however hazardous because there is not an exhaustive control of the status of the ZMP during the double support phase, and that is a potential risk. A similar development described in [107] solves that problem.

Another solution, proposed by Kajita et al. in [48, 47], to avoid this problem is to employ flexible step duration and step sizes. According to the 3D-LIPM model, given an initial position $x_i^{(n)}$ and velocity $v_i^{(n)}$ at the beginning of the n-th step, the final position $x_f^{(n)}$ and velocity $v_f^{(n)}$ of the CoM are given by:

$$\begin{pmatrix} x_f^{(n)} \\ v_f^{(n)} \end{pmatrix} = \begin{pmatrix} C_T & T_c S_T \\ S_T/T_e & C_T \end{pmatrix} \begin{pmatrix} x_i^{(n)} \\ v_i^{(n)} \end{pmatrix} \tag{2.41}$$

where $C_T \equiv \cosh(T_S/T_C)$, $S_T \equiv \sinh(T_S/T_C)$, $Tc \equiv \sqrt{z_c/g}$, z_c is the height of the CoM, g is the gravitational constant and T_s is the duration of the single support phase.

Nevertheless, as mentioned before, there exist only a possible trajectory for the CoM given an initial state and reference location for the pivot point, hence it is not possible to specify the final velocity and position of the CoM at the same time. The solution proposed by Kajita et al. is to set a weight a to the desired position x_d and a weight b to the desired velocity v_d and minimize the norm of the error:

$$N \equiv a(x_d - x_f^{(2)})^2 + b(v_d - v_f^{(2)})^2 \tag{2.42}$$

to obtain a feasible combination of x_f and v_f that is close to the target state. Substituting (2.41) in the previous expression it is possible to obtain the value of $x_i^{(2)}$ that minimizes the norm of the error:

$$x_i^{(2)} = \frac{\left(aC_T(x_d - S_T T_c v_d)^2 + b\frac{S_T}{T_c}(v_d - C_T v_f)\right)}{aC_T^2 + b(\frac{S_T}{T_c})^2}. \tag{2.43}$$

In this way, by setting the proper step size, it is possible to achieve an state of the CoM at the end of the step that minimizes the norm of the error in position and velocity for a target state. Moreover, in case a double support phase is needed, it is only necessary to increase the size of the step by the amount covered during the double support period, T_{dbl}, at the velocity reached at the end of the previous single support phase:

$$d = v_f^{(1)} T_{dbl}. \tag{2.44}$$

2.5.2.3 Preview Control with 3D-LIPM

The equations that describe the movement of the CoM according to the 3D-LIPM can be viewed from a different perspective, the cart table-model. Figure 2.7 depicts this model. A running cart of mass m runs on a pedestal table whose mass is negligible. Since the sagittal and lateral components of the movement are independent in the 3D-LIPM, the equations resulting from the application of this model can describe the dynamics of any of the components. The intuitive idea that this model helps understanding is that the cart has to accelerate at a certain rate towards the outer end of the table to prevent it from rotating. If the cart manages to keep the table upright, the torque around the edge of the pedestal is zero:

$$\tau_{zmp} = mg(x - p_x) - m\ddot{x}z_c = 0. \tag{2.45}$$

The position of the ZMP depends therefore on the position of the cart and on its acceleration. By defining the time derivative of the sagittal acceleration of the CoM as:

$$\frac{d}{dt}\ddot{x} = u_x \tag{2.46}$$

the equations of the dynamical system can be written as:

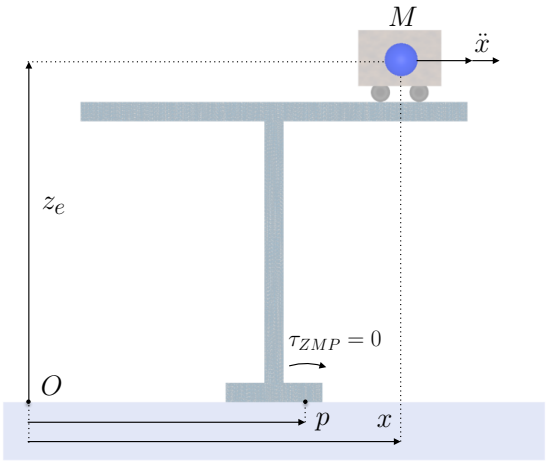


Figure 2.7: Cart-Table model

$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_x \tag{2.47}$$

$$p_x = \begin{bmatrix} 1 & 0 & z_c/g \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} \tag{2.48}$$

The problem of generating a trajectory for the CoM that matches a reference trajectory for the ZMP can be solved with servo control techniques. Kajita et al. proposed in [45] to employ a preview control approach to obtain an optimized solution to this problem. Figure 2.8 illustrates the configuration of a servo-controller that changes the first derivative of the acceleration (jerk) of the CoM to track the reference ZMP, p^{ref}. Nevertheless, the use of ordinary servo-control involves a certain delay in the tracking of the ZMP trajectory, Figure 2.9a, that is not acceptable for walking pattern generation. Since future information for the reference ZMP is available, it is possible to employ techniques that produce a much tighter match of the ZMP signal, Figure 2.9b.

In order to apply preview control to the ZMP of the robot, we first define a state of the system $\boldsymbol{x}(k) \equiv [x, \dot{x}, \ddot{x}]^T$ composed of the position, velocity and acceleration of the CoM in the sagittal plane. The procedure is identical in the frontal plane.

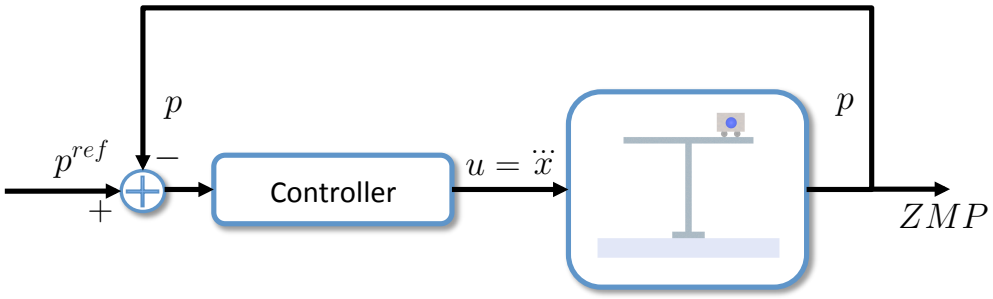


Figure 2.8: Servo configuration for ZMP tracking.

The first step to apply preview control is to discretize the system with sampling period T: $\boldsymbol{x}(k) \equiv [x(kT), \dot{x}(kT), \ddot{x}(kT)]^T$. Additionally, the reference position for the ZMP in the sagittal plane is $p(k) \equiv p_x(kT)$, and the control action applied to the system is: $u(k) \equiv u_x(kT) \equiv \dddot{x}(kT)$. In this way, the state equation of the system is given by:

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}u(k), \tag{2.49}$$

$$\boldsymbol{p}(k) = \boldsymbol{C}\boldsymbol{x}(k), \tag{2.50}$$

where

$$\boldsymbol{A} \equiv \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.51}$$

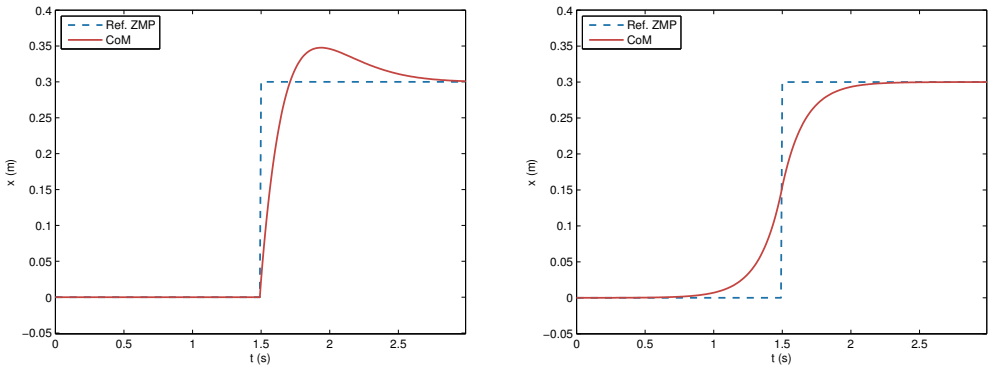
$$\boldsymbol{B} \equiv \begin{bmatrix} T^3/6 \\ T^2/2 \\ T \end{bmatrix}, \tag{2.52}$$

$$\boldsymbol{C} \equiv \begin{bmatrix} 1 & 0 & -z_c/g \end{bmatrix}. \tag{2.53}$$

Then, in order to obtain an optimized value of the control action, the following performance index is defined:

$$J = \sum_{i=k}^{\infty} \left\{ Q_e e(i)^2 + \Delta e \boldsymbol{x}^T(i) Q_x \Delta e \boldsymbol{x}(i) + R\Delta u^2(i)) \right\}. \tag{2.54}$$

where $Q_e, R > 0$ and Q_x is a 3×3 symmetric non-negative definite matrix.All Q_e, R and Q_x are weights parameters of the performance index. On the other hand, $e(i) \equiv p(i) - p^{ref}(i)$ is the servo error.



(a) Ordinal Servo Control (b) Preview Control

Figure 2.9: Preview Control approaches take advantage of future information about the reference trajectory.

By minimizing the performance index J over the preview interval, the gains G_i, G_x and G_p that determine the proper control signal are determined. In this way, the control signal becomes:

$$u(k) = -G_i \sum_{i=0}^{k} e(k) - G_x \boldsymbol{x}(k) - \sum_{j=1}^{N_L} G_p(j) p^{ref}(k+j). \tag{2.55}$$

The use of preview control of the ZMP for balance control in humanoid robots has been successfully applied in different configurations since it was first proposed. In this way, Wieber proposes an efficient implementation of the approach in [105]. In [17, 18] the work is further developed, showing the use of a solver for the quadratic program and proposing the definition o irregular sampling periods. In [77], the preview control is also employed to generate walking patterns online, and in [93] an implementation on a soccer robot is described.

2.5.3 Passive Dynamic Walking

For several centuries there has been a type of toy with the shape of a human body that is able to walk down a slope. The toy alternates its support from one foot to the other and does not require any energy source other than gravity. The gait style displayed by these devices, which are called passive dynamic walkers, resembles remarkably that of human beings. The efficient use of energy in both approaches is probably the cause of their resemblance.

Passive dynamic walkers are able to trade the potential energy that they have at the upper part of the slope with the energy lost in friction and collisions with

the floor. Moreover, given a certain passive walker, adding a source of energy that allows the robot to walk in a horizontal floor only requires minor modifications [67].

Since one of the potential applications of research in humanoid locomotion is the development of leg prosthesis that provide motion capabilities to disabled patients, the understanding of the underlying mechanics of passive dynamics walkers is important both because of the reduced energy consumption and because of the similarity to human movements.

In this way, several works have deepened into the mechanics of passive walkers. First, in [68] the interest of the problem was highlighted and the first studies on the dynamics of passive dynamic walking were introduced, but the motion were constrained only to the sagittal plane. In [22] the understanding of the dynamics is augmented, [60] extends the movement to include lateral tilting and in [61] the use of power sources is incorporated. In [62] the energetic costs of walking are analyzed. A complete description of the construction procedure of a dynamic passive walking system is included in [106], where the resistance against disturbances is also evaluated.

One of the problems of passive dynamic walkers is the difficulty to execute complex motions. Since the hardware design of passive dynamic walkers is oriented to optimize the energy consumption during the gait, it is not always possible to perform different tasks. Hence the popularity of passive dynamic walking approaches has been reduced compared to that of Zero Moment Point approaches.

2.5.4 Momenta Tracking

The use of the ZMP criterion for walking pattern generation aims at preventing the appearance of torques around the edges of the sole that will cause the robot to tip over. Assumed that the state of the robot is balanced, this is a convenient strategy to generate stable motions. However, under some circumstances, it will be useful to apply torques of a controlled magnitude. For instance, if the robot is significantly perturbed, it will rotate around the edges of the soles. In that case, the size of the support polygon is so small that the equations of the ZMP are hardly applicable.

Moreover, even though simplified models such as the 3D-LIPM enjoy a great success, they neglect the rotational inertia of the parts of the robot, which plays a fundamental role in biped balance as has been pointed out in [85]. To cope with this problem, the previously described versions of the inverted pendulum that account for angular momentum have been proposed.

A more general approach to ensure balance, however, is to directly control the centroidal angular momentum of the system. In the same way, an alternative

procedure to the use of reference ZMP trajectories to generate motions in a determined direction is the control of the linear momentum of the robot. That is equivalent to control the velocity of the CoM of the robot. The combination of reference angular and linear momentum tracking is therefore a general approach to define whole-body balanced motions for the robot.

The challenge of obtaining a set of equation that allow to generate the walking pattern given reference angular and linear momenta was taken by Kajita et al. in [46], where they propose the resolved momentum control approach, whose results are briefly summarized next. The linear P and angular L momenta of the whole robot are given by:

$$\begin{bmatrix} \boldsymbol{P} \\ \boldsymbol{L} \end{bmatrix} = \begin{bmatrix} \tilde{m}\boldsymbol{E} & -\tilde{m}\hat{\boldsymbol{r}}_{B\to\tilde{c}} & \boldsymbol{M}_{\dot{\boldsymbol{\theta}}} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{H}_{\dot{\boldsymbol{\theta}}} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_B \\ \boldsymbol{\omega}_B \\ \dot{\boldsymbol{\theta}} \end{bmatrix} \tag{2.56}$$

where $\tilde{m}$ is the mass of the robot, E is an identity matrix, $\hat{\boldsymbol{r}}_{B\to\tilde{c}}$ is the vector from the base link to the CoM, $\boldsymbol{I}$ is the inertia matrix with respect to the CoM and $\boldsymbol{M}_{\dot{\boldsymbol{\theta}}}$ and $\boldsymbol{H}_{\dot{\boldsymbol{\theta}}}$ are the inertia matrixes that indicate the effect of the joint velocity in linear and angular momentum respectively. On the other hand, v_B and ω_B are the linear and angular velocity of the base of the robot respectively and $\boldsymbol{\theta}$ is a vector with the angular velocity of all joints of the robot.

By following the procedure described in [46] it is possible to obtain the following expression for the motion of the base link and the rotational velocity of the joints of the robot:

$$\begin{bmatrix} \boldsymbol{\xi}_B \\ \dot{\boldsymbol{\theta}}_{free} \end{bmatrix} = \boldsymbol{A}^{\dagger}\boldsymbol{y} + (\boldsymbol{E} - \boldsymbol{A}^{\dagger}\boldsymbol{A}) \begin{bmatrix} \boldsymbol{\xi}_B^{ref} \\ \dot{\boldsymbol{\theta}}_{free}^{ref} \end{bmatrix} \tag{2.57}$$

with

$$\boldsymbol{y} \equiv \boldsymbol{S}\left\{ \begin{bmatrix} \boldsymbol{P}^{ref} \\ \boldsymbol{L}^{ref} \end{bmatrix} - \sum_{n=1}^{2} \begin{bmatrix} \boldsymbol{M}_{Fi}^{*} \\ \boldsymbol{H}_{Fi}^{*} \end{bmatrix} \boldsymbol{\xi}_{Fi}^{ref} \right\} \tag{2.58}$$

$$\boldsymbol{A} \equiv \boldsymbol{S}\begin{bmatrix} \boldsymbol{M}_B^{*} & \boldsymbol{M}_{free}\boldsymbol{H}_B^{*} & \boldsymbol{H}_{free} \end{bmatrix} \tag{2.59}$$

where $\boldsymbol{\xi}$ contains the linear and angular velocities and $\boldsymbol{S}$ is a selection matrix that allows to specify the elements of the angular and linear momenta to be controlled. Subindex B stands for the base link, F_i is the frame located at the leg i, the links whose position can be modified are marked with $free$. The superindex ref, on the other hand, denotes reference values to be tracked. Finally, $\boldsymbol{A}^{\dagger}$ is the pseudo-inverse of A.

This general solution to the problem of angular and linear momentum control given a set of contact restriction provides an excellent tool to design balanced

motions for the robot. However, its use in physical robots is hindered by the computational requirements of the approach. Additionally, the approach can be affected by numerical instabilities. For this reason, several authors have proposed alternative solutions to the problem of linear or angular momentum control.

Goswami et al. describe in [24] the command of a humanoid robot through the control of the rate of change of linear and angular momentum. The approach is extended in [64] to include the use of steps in the control of balance. In [2], on the other hand, the motion-embedded CoM Jacobian framework [53, 13] is modified to utilize angular momentum control.

Finally, the control of angular momentum has also offered interesting results in the animation domain. In [66] the centroidal angular momentum is considered to obtain a set of control laws for the CoP while in [39] is introduced a system that prioritizes the control of angular or linear momentum according to the circumstances to obtain biologically realistic motions.

2.6 Balance Control

In the previous section we described several approaches to generate the walking pattern of a biped robot. Starting from the state of the system at a certain instant it is possible, in most of those approaches, to generate the future poses of the robot. If one assumes that the state of the system is simply the state scheduled by the walking pattern generator in a previous cycle, it is possible to generate a walking pattern without any feedback from the sensors. In this case, there is no control stage, and the robot operates in open-loop mode.

Although walking pattern generators in open-loop mode are easy to implement and numerically robust, they do not correct differences between the model and the real robot and do not adapt to changes in the environment or perturbations. Hence it is important to use the readings of the sensor as feedback for the walking pattern generator. If the sensor feedback is employed to create a signal that is mixed with an open-loop walking pattern generator, the tasks of planning and control are detached. On the other hand, if the sensor feedback updates the internal state of the system of the walking pattern generator, the planning and control activities execute in an unified way.

As we will see below, there is a wide range of controllers that have been proposed to control the balance of robots. It might seem difficult to select the best performing controller for a given platform. However, since the controllers usually aim at improving a particular source of instability, it is frequent to integrate different balance controllers in the system. A paradigmatic combination of controllers can be found in the state-of-the-art robot ASIMO [99].

2.6.1 Detached Planning and Control

Planning and control have been traditionally separated stages in robotics. It is a common practice in industrial plants to apply PID controllers to the actuators in order to ensure the accurate tracking of reference trajectories. It is natural, therefore, that some of the first attempts to include feedback in walking pattern generators of biped robots followed the same strategy [37].

Among the deviations of the trajectories of the poses of the robot from their reference trajectories, the most critical situation for balance happens when the robot rotates around the edges of the soles. In that situations, the feet are likely to hit the ground while performing a step. Moreover, the plane of movement of the CoM gets inclined, which changes the direction of the inertial forces generated by the walking patter generator and causes unexpected torques. For that reason, efforts of the controllers are usually addressed to maintain the upright position of the trunk.

Additionally, the separation between planning and control was also motivated in the early developments by the scarce computing resources. In [97] the location of the CoM is modified online to improve the tracking of the ZMP proyected by an offline generated walking pattern. In [94] the CoM trajectory generated by an open-loop walking pattern generator is manipulated with whole-body cooperation using the CoM Jacobian. Much simpler PID schemes have also been successfully applied in various works. In this way, Faber et al. [19] proposed the use of P controllers that act on the ankle joints according to the gyrometer readings. Similarly, [54] employed PI controllers for the ankle joints to prevent the robot from tilting over. In [75] the type of controller employed is PID and the control action is exerted on the hip joints to stabilize the balance of the robot after ball kicks. On the other hand, an original point of view is taken by Kuinderma et al. in [58]. In that work, Bayesian optimization is employed to obtain arms movement policies that minimize the total energy expenditure and the balance recovery. Finally, the control of the stiffness of the limbs to improve the balance is pointed out in [78, 73].

2.6.2 Unified Planning and Control

The use of sensor readings to modify the internal variables of the walking pattern generator has become increasingly popular in recent works. The reason is that, in certain cases, it involves important advantages with respect to the detached control strategy. For instance, if the inverted pendulum model is employed to generate the trajectories of the CoM, the use of a PID controller to track the target trajectory will generate undesired forces that invalidate the calculations obtained through the application of the inverted pendulum model. On the other

hand, if the state of the CoM estimated by the sensor readings is employed to update the internal state of the walking pattern generator, the model will still be perfectly valid and the state will reproduce a situation which is closer to reality. The sensors feedback can be applied to different elements of the system, as explained next.

2.6.2.1 CoM Manipulation

The manipulation of the CoM dynamics scheduled by the walking pattern generator is a typical activity to control the balance in inverted pendulum approaches. However, the strategy adopted to alter the dynamics of the CoM depend on the perspective employed.

A first group of approaches simply aim at maintaining an internal CoM state that is close to reality. In this way, the sensor readings are processed to estimate position, velocity or acceleration of the CoM and the resulting estimates are fused with the internal state of the walking pattern generator. Some examples of this type of feedback strategy are [15, 16, 29, 30], which include formally the concept of CoM observers and [25] that updates the location of the CoM (approximated by the trunk) in case it exceeds a certain threshold.

In the second group of approaches, the control signals are applied to the CoM, in an indirect way, by adapting the trajectory of the ZMP. In this manner, in [82] the CoM is moved vertically to correct the deviation in the location of the ZMP. Likewise, in [49] the output of a PD controller for the torso attitude is combined with the output of a P controller that takes the error in the location of the ZMP. The resulting signal is employed to manipulate the reference trajectory of the ZMP, improving the accuracy in the location of the ZMP projected by the whole body model at a reduced computing cost. In a similar way, the successive versions of the model ZMP control of ASIMO manipulate the location of the ZMP to correct the deviation in the body attitude [37, 98, 99].

2.6.2.2 Torso Attitude

There is another group of approaches that aim at improving the balance control of a robot, usually after a perturbing event, by bending the hip of the robot. This behavior, known as hip strategy [5], has also been observed in humans. The physical explanation of this reflex is that, by bending the hip, the mass of the upper part of the body generates and angular momentum that helps maintaining the horizontal position of the soles.

In the work presented by Abdallah et al. [1], disturbance response is divided in two stages. First, the hip is bent to absorb the perturbation, and second, the torso slowly recovers its upright pose. The strategy to recover the original pose

is based on the maximization of the potential energy of the body.

The sophisticated scheme introduced by Kajita et al. for running robots, [50], includes a complex combination of different types of controllers. The control of the angular momentum is managed by a resolved momentum controller. To recover the reference attitude of the body, a PD controller is used to set the attitude of the torso.

The hip strategy is also employed in the works by Yi et al. [111, 109, 110], among other biomechanically motivated strategies. In these works, the parameters of the control laws of the balance controllers are learned.

2.6.2.3 Stepping

The adaptation of the landing position of a foot and the duration of the step are paramount tools in the balance control of legged robots. Let the amazing performance of the stepping controllers of Boston Dynamic's Big Dog quadruped robot be a proof of it [88]. Nevertheless, stepping also involves a risk. A rapidly executed step may lead to an uncontrolled impact between the extremity and the ground. If the leg is not able to absorb the impact, it will generate a destabilizing torque. For this reason, compliant controlled robots are more suitable for extreme stepping actions.

However, abundant stepping controllers addressed to the, more common, position controlled robots have been proposed. For instance, in [41] an offline generated ZMP-based walking pattern is combined with real time modification of landing position. The ZMP stability criterion plays an important role in other stepping controllers, like [101, 72], where both the duration and position of the step are adapted according to the circumstances to allow the ZMP to remain within the support polygon. The analytic solution proposed in that work is combined with a solver that plans CoM and ZMP simultaneously in [74]. The problems derived from unexpected collisions or the presence of holes in the ground is studied in [73]. In [112], on the other hand, the step size is calculated to maintain the angular momentum of the system at a reduced level. In [12] a stepping controller for hopping robots is described and in [70] the concept of orbital energy of the CoM is studied to adapt the size and duration of the steps in the frontal plane. Finally, the application of a stepping controller to animated avatars is explained in [14].

2.7 Motion Perception

Providing quality information to the balance controllers is essential for their correct performance. In this section we review the type of sensors that are

related to motion tasks. However, in most cases, the information required by the walk engine is not directly provided by the sensors or the readings are noisy. Additionally, redundant information is also common in humanoid sensors. Hence the issues of sensor processing and fusion need to be considered.

2.7.1 Motion Sensors

Humanoid robots are usually equipped with a wide range of sensors. Depending on whether the sensors measure information about the state of the robot or about its surrounding environment they can be categorized into propioceptive or exteroceptive sensors.

Typical propioceptive sensors include thermometers, voltmeters for the battery state, joint position sensors, accelerometers, gyrometers, compasses, force sensors, etc. Among them, the most important for balance control are joint position sensors, gyrometers, accelerometers and force sensors.

Joint position sensors are a key component of servomotors, and are therefore a mandatory element in position controlled robots. Apart from being necessary to set the joints at a target position, they also provide a measure of the error in the performance of the joint actuator. In this way, it is possible for the balance controllers to compute the difference between the pose commanded to the robot and the actual pose to improve the estimation of the internal state of the system.

Accelerometers provide a mean to detect disturbances and the direction of gravity. The acceleration of the CoM is part of the internal state of some walk engines. Additionally, they serve to trigger reflex motions after severe perturbations and prevent the robot from damaging itself. The estimation of the direction of gravity, by its side, is of paramount importance to estimate the attitude of the torso. However, since gravity acceleration is mixed with the acceleration caused by perturbations or by the walk engine itself, its magnitude is mainly of importance in the long-term. Accelerometers alone cannot be used to estimate abrupt changes in the attitude of the torso.

Gyrometers are therefore the perfect complement to accelerometers. By integrating rotational velocity estimations, it is possible to obtain accurate estimations of the changes in the attitude of the torso. The drawback of the integration of readings is that errors are cumulative. Hence the readings of gyrometers and accelerometers must be combined to deliver rapid and accurate estimations of the robot attitude without a long-term drift.

The estimation of the reaction force exerted by the ground on the robot provides also practical information for balance control. As mentioned before, the center of pressure matches the location of the ZMP in balanced gaits. Moreover, the proximity of the center of pressure to the edge of the sole provides a measure of the stability margin.

By the other hand, exteroceptive sensors of humanoid robots usually include microphones, cameras and ultrasonic, infrared or laser obstacle detectors. Among them, the most useful for balance control are cameras. In this manner, in [6] is shown how balance in humans deteriorates with closed eyes. However, visual information is complex and requires considerable computing resources, which hiders their use for balance control. Finally, obstacle detectors allow robots to prevent collisions, hence the information that they provide is used for path planning rather than for balance control.

2.7.2 Sensor fusion

The information provided by the humanoid sensors needs to be processed previously to its use by the feedback controllers. Notably, redundant information and noise of the readings need to be considered. Combining redundant information can be easily achieved with a weighted sum of the readings. On the other hand, a simple first order low-pass filter can be used to reduce the noise and prepare the sensor information to its use by the balance controllers. Nevertheless sophisticated sensor fusion approaches such as the Kalman Filter [52] provide better results. The approach is described in [8] as follows:

> *"The Kalman filter is a set of mathematical equations that provides an efficient computational (recursive) means to estimate the state of a process, in a way that minimizes the mean of the squared error".*

Let us briefly explain the operation of the filter. The goal of the approach is to obtain an estimation of the state of the system, x_k, which is governed by the differential equation:

$$\boldsymbol{x}_k = \boldsymbol{A}\boldsymbol{x}_{k-1} + \boldsymbol{B}\boldsymbol{u}_{k-1} + \boldsymbol{w}_{k-1}, \tag{2.60}$$

being k is the discrete time index. Matix $\boldsymbol{A}$ is the state transition model of the system while $\boldsymbol{B}$ is the control-input model that is applied to input $\boldsymbol{u}_k$ and $\boldsymbol{w}$ is the process noise.

The set of sensor readings, $\boldsymbol{z}$, is related to the system state by matrix $\boldsymbol{H}$:

$$\boldsymbol{z}_k = \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{v}_k. \tag{2.61}$$

with the measurement noise represented by $\boldsymbol{v}_k$. Both $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ are assumed to be drawn from zero mean multivariate normal distributions, with respective covariances $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$:

$$p(\boldsymbol{w}) \sim N(0, \boldsymbol{Q}), \tag{2.62}$$
$$p(\boldsymbol{v}) \sim N(0, \boldsymbol{R}). \tag{2.63}$$

At this point, we make differences between *a priori* and *a posteriori* estimates of the state of the system, denoted by $\hat{\boldsymbol{x}}_k^-$ and $\hat{\boldsymbol{x}}$ respectively. In this way, the *a priori* and *a posteriori* estimation errors are:

$$\boldsymbol{e}_k^- \equiv \boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^-, \tag{2.64}$$
$$\boldsymbol{e}_k \equiv \boldsymbol{x}_k - \hat{\boldsymbol{x}}_k. \tag{2.65}$$

The corresponding *a priori* and *a posteriori* estimates of the covariance matrix are:

$$\boldsymbol{P}_k^- = E[\boldsymbol{e}_k^- \boldsymbol{e}_k^{-T}], \tag{2.66}$$
$$\boldsymbol{P}_k = E[\boldsymbol{e}_k^- \boldsymbol{e}_k^T]. \tag{2.67}$$

As explained in [8], it is possible to obtain an estimate of the state of the system that minimizes the *a posteriori* error covariance. For that, the filter is configured as a predictor-corrector algorithm. In this way, the approach firstly performs a predictor (or time update) stage where the state and error covariance estimates are projected in the future to obtain the *a priori* estimates of the next step:

$$\hat{\boldsymbol{x}}_k^- = \boldsymbol{A}\hat{\boldsymbol{x}}_{k-1} + \boldsymbol{B}\boldsymbol{u}_{k-1} \tag{2.68}$$
$$\boldsymbol{P}_k^- = \boldsymbol{A}\boldsymbol{P}_{k-1}\boldsymbol{A}^T + \boldsymbol{Q} \tag{2.69}$$

Secondly, the correction (or measurement update) stage processes the information provided by the measurements to improve the *a posteriori* estimates:

$$\boldsymbol{K}_k = \boldsymbol{P}_k^- \boldsymbol{H}^T (\boldsymbol{H}\boldsymbol{P}_k^- \boldsymbol{H}^T + \boldsymbol{R})^{-1} \tag{2.70}$$
$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k(\boldsymbol{z}_k - \boldsymbol{H}\hat{\boldsymbol{x}}_k^-) \tag{2.71}$$
$$\boldsymbol{P}_k = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H})\boldsymbol{P}_k^- \tag{2.72}$$

By alternating both stages the filter effectively achieves the minimization of the *a posteriori* error covariance. The innovation matrix, $(\boldsymbol{z}_k - \boldsymbol{H}\hat{\boldsymbol{x}}_k^-)$, provides an interesting insight into the performance of the filter, since it measures the discrepancy between the predicted measurements and the actual ones.

2.8 Summary

The chapter begins with an explanation of the potential applications of humanoid robots and subsequently reviews the literature regarding balance control.

Concerning its applications, humanoid robots provide a convenient interface to the increasingly smart software that are being developed. Since humans have modeled the environment and numerous tools to adapt to the shape of our body, a machine with a similar interface would benefit from excellent interacting capabilities. Moreover, the use of robotic parts either as a replacement or complement to human limbs has a paramount importance in the medicine and military fields.

However, the development of intelligent humanoid robots involves tremendous challenges in areas like perception, reasoning, natural language processing, creativity, knowledge representation and motion. Since this work focuses on the balance control of a biped gait, the relevant background was reviewed in this chapter.

In order to obtain a balanced gait, two groups of approaches for walking pattern generation in legged robots can be distinguished. The first one, which does not require any physical model of the robot, is based on central pattern generators. Approaches belonging to this group usually combine sets of oscillatory signals to create omnidirectional gaits. However, the procedure of tuning the parameters of the gait is quite elaborated and platform specific. The second group, on the other hand, uses simplified dynamic models of the robot and stability criteria to generate the balanced walking pattern.

Biped models proposed in the literature aim at attaining an elevated simplicity while maintaining the accurate representation of the fundamental features of motion dynamics. One of such models is the inverted pendulum, which states that all the mass of the robot is concentrated at the position where the CoM is located. The conceptual and mathematical simplicity of the inverted pendulum model caused it to be employed as a base for many other models. Notably, the introduction of the 3D Linear Inverted Pendulum Mode enabled to represent the non-linear dynamics of the CoM by means of second order linear differential equations. This opened the door to new types of approaches for walking pattern generation and balance control. However, the simplification involved by the use of the inverted pendulum is excessive in some cases. The main drawback of this simplifications is the neglect of the rotational inertia of the robot body. For that reason, several models have been proposed that sacrifice a certain level of simplicity of the inverted pendulum in exchange of a more accurate representation of the angular momentum.

Defined the model of the robot, it is necessary to set a stability criterion in order to generate the walking pattern. The simpler stability criterion, which can only be applied to static systems, requires the vertical projection of the CoM of the robot to lie within the boundaries of the support polygon. For moving devices, however, this criterion is highly inaccurate, since it neglects inertial forces. The inclusion of inertial forces gave rise to the most popular stability criterion, which

is based on the Zero Moment Point concept.

The ZMP specifies the point on the ground where the tipping moment acting on the robot, due to gravitational and inertial forces, equals zero. The criterion specifies that in order to obtain a balanced robot, there must exist a ZMP. If this point exists, it can only be contained in the limits of the convex hull of the support polygons. Otherwise, gravitational and inertial forces would induce a moment around an axis placed at the closest side of the convex hull. For this reason, the application of the ZMP to robots whose support polygon size is reduced is complicated. For instance, when a robot receives an external impact on its back, the perturbation can easily make it lose momentarily its balance and pivot around the toes. In that common situation, the ZMP criterion is of no use because the size of the support polygon is too small to project the ZMP inside. To cope with this problem, the Ground Rotation Indicator was proposed, providing a tool not only to ensure balance but also to measure instability. Finally, the more complex strong and weak stability criteria were introduced to account for three dimensional contacts between robot and environment and to consider friction.

Nevertheless, the simple and efficient ZMP concept has remained for long the most popular stability criterion for walking pattern generation. Since the calculation of the exact position of the ZMP is a complex and computationally demanding task, the use of simplified models has severe benefits. Particularly, the simplification of the dynamic model of the system performed by the 3D-LIPM was paramount in the implementation of the first ZMP-based online walking pattern generators. In those early implementations, the trajectories of the CoM were obtained analytically. Subsequent new approaches based on preview control theory allowed to obtain smoother CoM trajectories that minimized the tracking error of the ZMP. However, the neglect of the rotational inertia hindered their performance. Recently, momenta tracking approaches have been proposed to provide accurate control of lineal and rotational momenta. Nevertheless, the computing requirements of these approaches are more elevated.

The generation of a theoretically balanced walking pattern is rarely sufficient to obtain a robust gait on a real platform. In this way, the existence of a balance control stage that uses the information provided by the sensors to improve the balance is fundamental. The estimation of the torso attitude or the location of the center of pressure are of great interest for balance. However, the information provided by the sensors can rarely be straightforwardly fed to the balance controllers. Since sensors signals are jerky and redundant, their readings must be processed previously to their use by the balance controllers. For this purpose, the Kalman Filter constitutes an excellent tool.

Options to include feedback are abundant and can be combined. A first group of approaches run in parallel to the open-loop generated walking pattern. In this way, the controller acts on elements like the arms, the CoM or the ankle

to maintain the conditions of the environments close to those expected by the walking pattern. The second group of approaches, on the other hand, modifies the internal state or configuration of the walking pattern generator according to the feedback. The elements that can be manipulated are also multiple in this case: CoM, ZMP trajectory, step size or duration, torso attitude, etc.

Nevertheless, the implementation of these approaches in real-world environments is not always straightforward. Differences between simplified dynamic models and real robots, limited computing capabilities, difficulties to estimate the state of the system with noisy sensors and the existence of sensory-motor lags are challenging problems that must be considered.

Chapter 3

Analysis of the Nao Humanoid Robot

3.1 Introduction

The study of the special features of the robotic platform employed is paramount in the design of the gait. In this chapter we review the technical specifications of the humanoid Nao and analyze the performance of its sensors and actuators. We also detail the calculations required to solve the forward kinematics problem and propose a solution for the inverse kinematics of this robot.

3.2 Technical Specifications

The robotic platform employed for this work is the 2010 version of the commercial humanoid robot Nao (Figure 3.1), developed by the French company Aldebaran Robotics[26, 27].

The Nao robot is 57 cm tall and weighs 4.5 Kg. The body mass index of the robot is therefore 13.5 Kg/m^2, which is significantly inferior to the body mass index of other humanoids such as ASIMO 2011($28.4Kg/m^2$) or HOAP-3 ($25Kg/m^2$).

Concerning computing capabilities, the robot is equipped with a x86 AMD GEODE CPU motherboard with 256 Mb of SDRAM memory. An ARM7-60Mhz microcontroller manages communication with the module microcontrollers distributed along the body. These microcontrollers implement servo-control of the motors and preprocessing of inertial unit data. The control cycle for the position controlled actuators runs at 100 Hz. These actuators use Maxon^{TM}cordless brush DC motors and spur and planetary gears, which are made of plastic loaded with PTFE (Polytetrafluoroethylene) and carbon fiber.

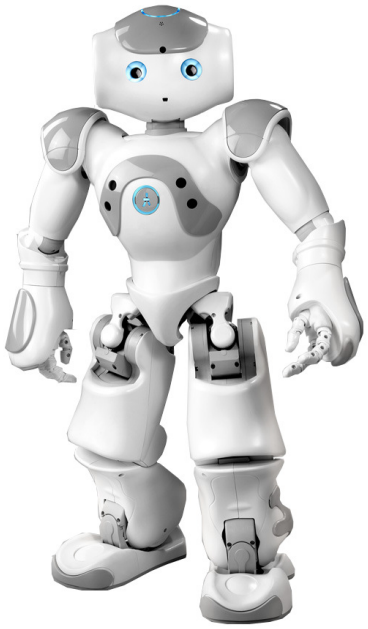


Figure 3.1: Aldebaran-Robotics Nao humanoid.

The robot is also equipped with an inertial unit that provides accelerometer and gyrometer measurements. According to the manufacturer specifications, the three-axis accelerometer has 1% precision in the range of $\pm 2G$ accelerations, and the two-axis (pitch and roll) gyrometer provides 5% precision for an angular speed range of $\pm 500^{\circ}/s$. In addition, eight Force Sensitive Resistors with a working range from 0 to 25 N are placed on the corners of the feet.

In the version used for this work, the robot has 21 degrees of freedom: five of them are in each leg, four are in each arm and two are in the neck. The remaining actuated joint is placed at one of the hips. It is 45 degrees inclined and it is coupled to the hip joint of the other leg, which is also 45 degrees inclined. The configuration of a shared joint between both legs is a special feature of the Nao robot that allows to save one motor compared to the typical 6 DoF per leg design of other humanoids. However, this innovative design complicates the inverse kinematics problem, because the pose of the torso and the swing foot cannot be fully specified simultaneously. The broad range of motions generated for this robot demonstrates that the sacrifice of one degree of freedom is a reasonable mean to reduce the cost of the robot.

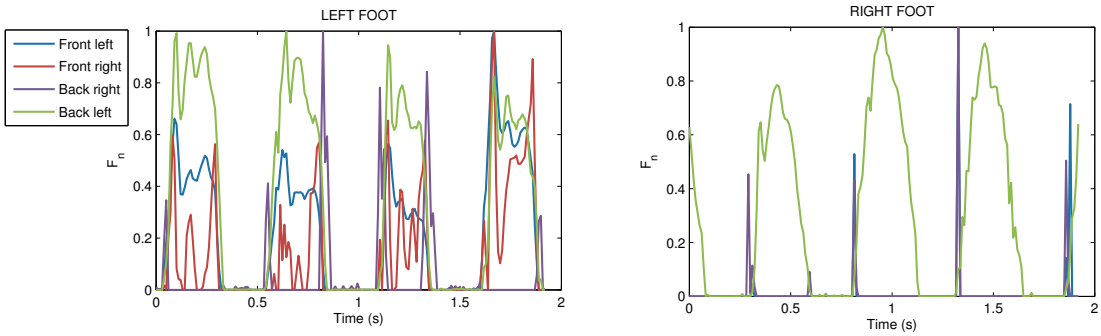


Figure 3.2: FSR raw signals.

3.3 Sensors Evaluation

The Nao humanoid is equipped with a rich set of sensors that can be exploited to improve balance. However, the noise level of some of these sensors is considerable. Hence it is necessary to evaluate the performance of each sensor in order to make the most of them.

3.3.1 Force Sensing Resistors

The Force Sensing Resistors (FSRs) are placed on corners of the soles of the feet. These sensors provide an estimation of the force applied at these points. Therefore, they could, ideally, be employed to determine the position of the center of pressure of the robot. In the case of Nao, on the other hand, this task is unworkable, due to the noise level and irregular performance exhibited by the sensors.

Figure 3.2 shows the readings of the FSRs of a Nao robot while stably walking on the spot. The signals have been normalized so that their maximum value along a 30s sequence is 1. In the plots of both legs, we can distinguish the alternation of the support and swing phases by the non-zero and zero regions of the normalized force, respectively.

The plot corresponding to the left foot shows the abrupt oscillations and the different typical patterns measured by each sensor. In the case of the back right corner, the signal only appears as peaks at the beginning and at the end of the support stages. This extreme behavior is more pronounced in the right foot, where only the signal of the back left sensor holds a high value during the whole support stage. Since similar problems of low performance of the FSRs also occur in other units, it might be due to the design of the sole of the robot.

Due to the irregular performance of the FSRs, its use as CoP estimators in this work was discarded. However, the event of a ground contact is much simpler

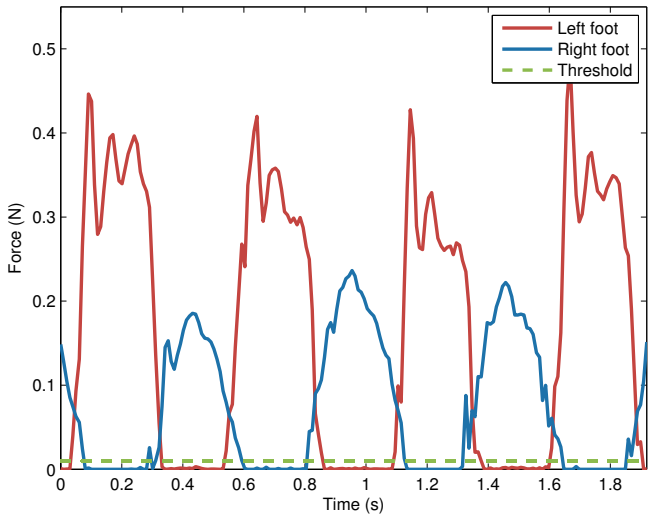


Figure 3.3: FSR raw signals grouped by foot.

to measure and it is also useful for balance. Figure 3.3 displays the total force measured by the set of FSRs available at each foot. In this way, we will employ a low force detection threshold to provide a simple and robust tool for ground contact detection.

3.3.2 Accelerometers

The Nao robot contains an inertial unit located at its chest that is composed of three accelerometers and two gyrometers. During the operation of the robot we find three different types of forces that can be measured by the accelerometers. The first one is the force of gravity, which has a high magnitude and a constant value pointing downwards. The second group consist of the inertia forces that apply on the torso due to the locomotion movements of the robot. Although their magnitude depends on the speed and smoothness of the gait, they are typically one order inferior to gravity. Finally, collisions with the ground, the self body or other objects also generate reaction forces. However, the flexibility of the body structure and the backlash of the joints dampen the magnitude of these forces and only strong impacts can be effectively detected.

According to the manufacturer data-sheets the accuracy of the 3-axis accelerometer unit is 1% in the range of 2G. Where G is the gravitational constant. However, we found that the most important limitation for these sensors was their sensitivity, which is in this model $0.175 m/s^2$. The significant quantization step size that causes this problem can be observed in Figure 3.4, which displays the

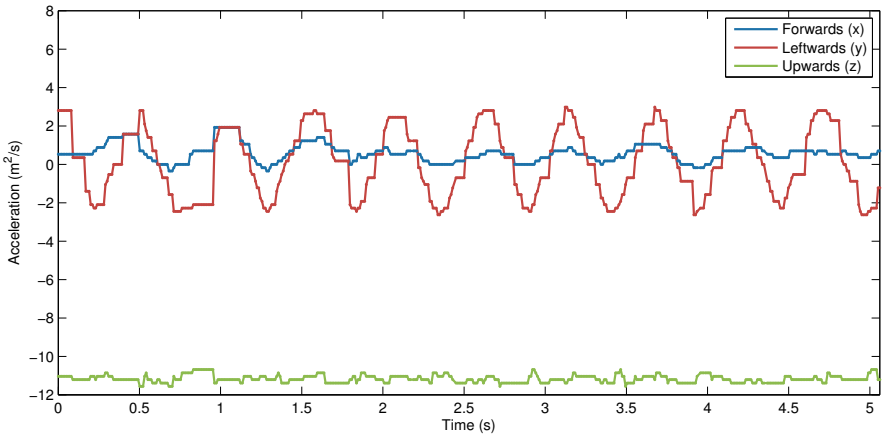


Figure 3.4: Acceleromters raw signals.

raw signals delivered by the accelerometers during a stable walking on the spot performance. This Figure also evidences the need for offset and gain tuning, because the signal of the x-axis accelerometer is positive in average and the acceleration measured in the z axis is higher than gravity.

Since gravity is commonly the main force that can be measured at the torso, the signals provided by the accelerometers can be employed to estimate its direction. By neglecting all other forces acting on the torso, we can estimate the roll and pitch angles of the torso in a very simple way:

$$pitch = \arcsin(\frac{a_x}{G}), \tag{3.1}$$

$$roll = -\arcsin(\frac{a_y}{G}), \tag{3.2}$$

where a_x and a_y are the acceleration in the x and y axis respectively and G the gravitational constant. However, if we apply the quantization step size to this formula we find that the minimum angle that can be detected with this approach is 1.02 degrees. Figure 3.5 shows the result of the use of the previous raw signals to estimate the torso orientation. The low resolution of the unit and the measurement errors introduced by other forces discard therefore the use of this approach of torso attitude estimation for accurate feedback of balance control. On the other hand, the main advantage of the approach is that the error in the estimation is not cumulative. It can be used thus to provide a rough initial estimate at boot time and to correct the drift in the estimation provided by the gyrometers, which we describe next.

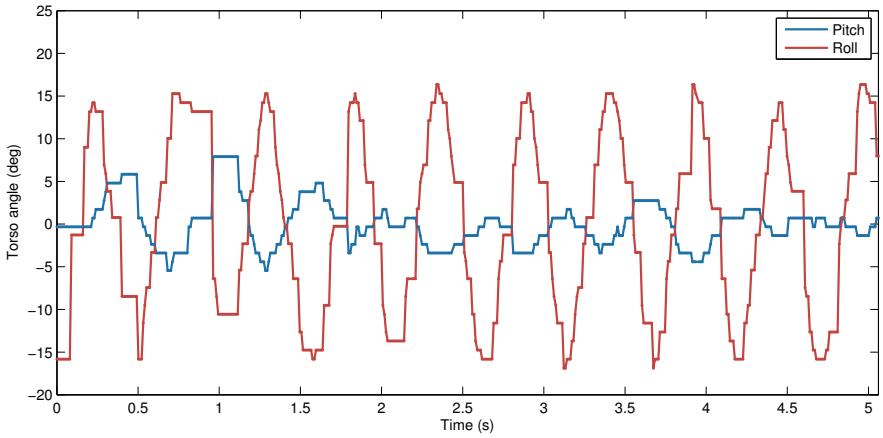


Figure 3.5: Torso angle estimation from gravity force direction.

3.3.3 Gyrometers

The gyrometer sensors located at the torso of the robot allow to estimate the rotational velocity at this point of the robot. Figure 3.6 displays the raw signals delivered by the gyrometers once again during a walking on the spot performance. Offset and gain tuning is also required by this sensors. According to the manufacturer, they provide and accuracy of 5% in the range of 500 deg/s. Although the rotational speed is an important feature for balance, we will concentrate on the attitude estimation that can be obtained through the integration of consecutive readings. While for low rotational velocities, the resolution in the estimation of the torso orientation is much higher than the estimation provided by the accelerometers, the drawback of this approach is that due to the integration procedure, the errors are cumulative and the estimations diverge after a few seconds. The drift in the estimation can be observed in Figure 3.7, which depicts the torso angle estimation obtained from the integration of the gyrometer measurements.

3.4 Actuator Evaluation

Differences between the performance of actual robots and simplified physical models depend on both the inaccurate readings provided by the sensors and the imperfect response of the actuators. In this section we analyze the delay and accuracy of the joint actuators.

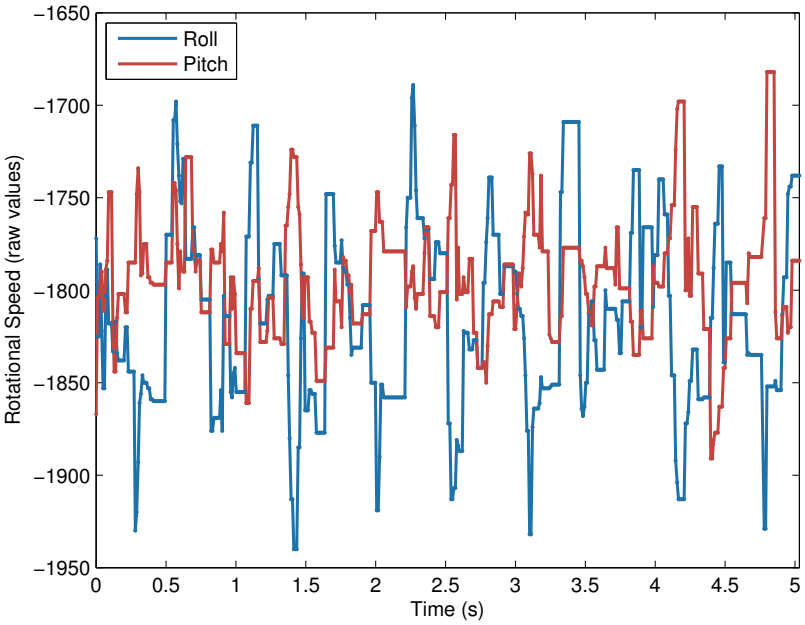


Figure 3.6: Gyrometers raw signals.

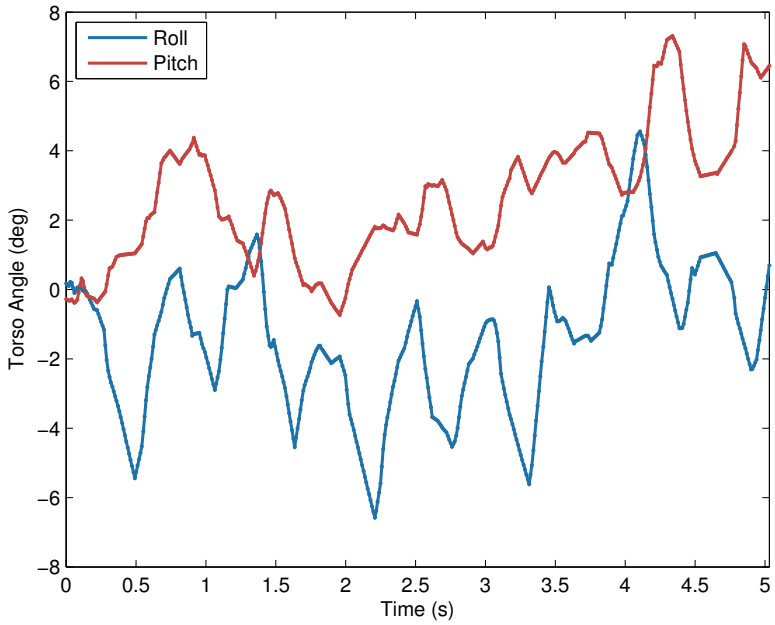


Figure 3.7: Torso angle estimation from integration of rotational velocity.

3.4.1 Sensory-motor Lag

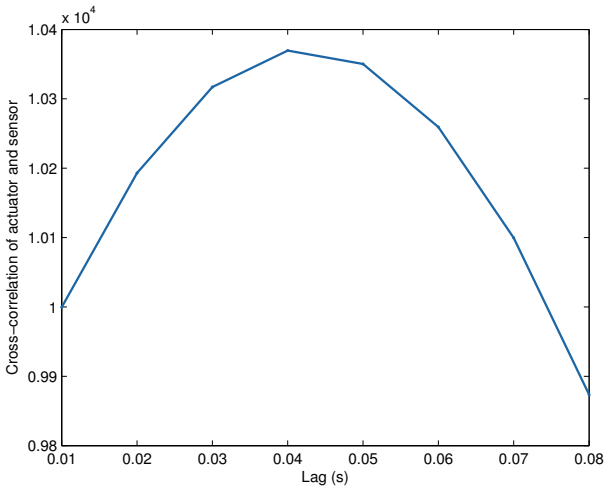


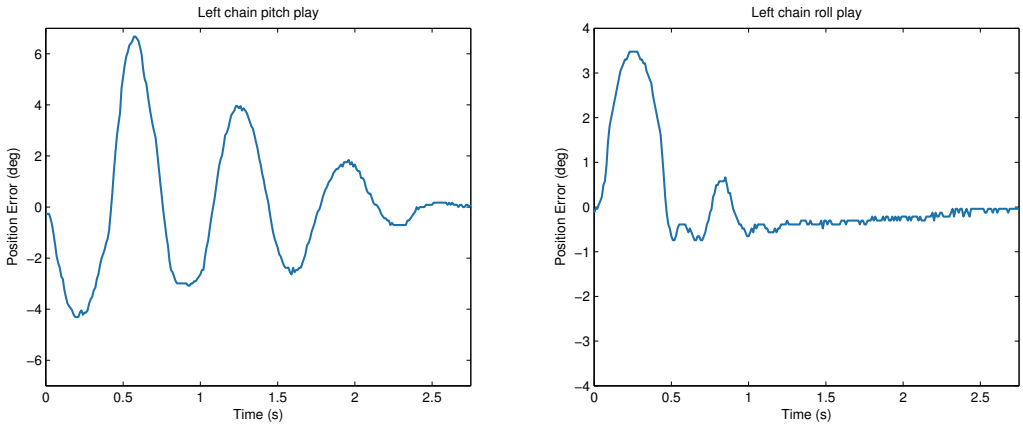
Figure 3.8: Estimation of the sensorimotor lag for the Nao robot.

The sensorimotor lag of the platform is a key point for the feedback controllers proposed in this work. Since this information is not provided by the manufacturer, it must be estimated. This is achieved by sending a 2 Hz sinusoidal signal to one of the actuators and then calculating the cross-correlation between that signal and the reading of the joint sensor. The samples of both signals are recorded at every control cycle before the updating operations of the middleware. Figure 3.8 shows the result of the cross-correlation. The maximum value of this function corresponds to a delay of 40 ms between the signal commanded to the motor and the signal read by the sensor.

3.4.2 Play of the joints

The accuracy in the execution of the movements commanded to the robot depend mainly on three factors: backlash of the gears, stiffness of the joints, and flexibility of the body parts. Since the body parts of the Nao platform are made of a relatively rigid plastic material we will focus in the two additional factors that determinate the accuracy of the movements.

The backlash of the gears is the maximum clearance between mated gear teeth. When the force applied at the end of a transmission chain reverses, each of the gears that compose the transmission chain have to move to cover the clearance. This effect results in a uncontrolled movement at the end of the chain equivalent to the sum of the clearance that all the gears have to cover. It is obviously an undesired effect. First because of its contribution to inaccurate movements of



(a) (b)

Figure 3.9: Play of pitch (a) and roll (b) in the kinematic chain of the left leg when the robot is pushed.

the body of the robot. And second, because the free movement of the gears ends usually abruptly, which adds additional forces to the system.

The stiffness of a servo is a measure of its resistance to deviate from its reference position under the action of a force. The advantage of having stiff joints is that they can track accurately the trajectories with a certain load. The opposite attribute is compliance. Compliant joints allow smoother movements, and are good at absorbing external (due to collisions) or internal (due to backlash) disturbances. A low level of stiffness is indeed preferred by some authors to improve balance [59]. Moreover, the stiffness of a joint is directly related to the torque that the motors deliver. Therefore, stiff joints involve a higher energy consumption and require more expensive motors.

Since the low-level API of the Nao robot allows the modification of a parameter related with the stiffness of each joint, a trade-off must be found between accurate movements and smooth trajectories with low energy consumption. For our locomotion system, we found that 0.7 was an optimal choice for legs and 0.3 for the head and the arms. The reason for such a low value in the head and the arms is double. First, they do not need to hold any weight during normal operation of the robot, and second, they are more elevated and thus exposed to greater damage in the event of impact against the ground.

Figure 3.9 illustrates the play of gears of the kinematic chains of the left leg. In 3.9b, the robot is pushed in his chest towards its back while standing still. The plot represents the error in the pitch angle of the torso according to the measures of the joint sensors of the left leg. In Figure 3.9a, a similar experiment is performed, but this time the robot is placed with its left foot in a step and

the right one in the air. Then, the robot is pushed laterally towards the left and the error in the roll angle of the torso is captured. The plots show a considerable amount of 10 degrees of uncontrolled movement in the case of the pitch angle of the torso and of 4 degrees in the case of the roll angle.

3.5 Forward Kinematics

The poses of the robot can be expressed as a function of a set of parameters. When the set of parameters employed is the array of joint positions, we say that the pose is configured in the joint space. The low level API of the Nao robot requires new reference positions for the 22 joints of the robot every 10 ms. However, the joint space is not the most convenient to plan the movements of the robot. In order to specify the motions of the robot, we combine the movement of three end-effectors. The first one is the pose of the foot that is not responsible for sustaining the robot (swing foot), the second one is the position of the Center of Mass, and the third one is the orientation of the torso. We will name the set of these three end-effectors the com-torso-foot space. In this way, the forward kinematics approach is the function that translates the coordinates of the joint space to the com-torso-foot space.

In the com-torso-foot space, the reference frame depends on the foot that is employed to sustain the robot (support foot). The origin of coordinates is always located at the intersection between sole plane and the line, normal to that plane, which goes through the ankle. This frame is called the "sole frame". The direction of the x axis points towards the toe of the foot, the y axis points to the external (left foot) or internal (right foot) side and the z axis is perpendicular to the sole plane and points upwards.

In the com-torso-foot space, the pose of the swing foot is expressed in Cartesian coordinates and roll-pitch-yaw angles, being those angles the rotations around the x, y and z axis respectively. The pose of the swing foot has therefore 6 dimensions. The position of the CoM, on the other hand, only requires three dimensions: the Cartesian coordinates of the point. Finally, the roll-pitch-yaw angles for the orientation of the torso provide three additional dimensions.

To express the transformations between each link of the kinematic chains of the robot we employ the Denavit-Hartenberg convention. By following the kinematic chains of the legs, arms and head starting at the torso, we obtain the homogeneous matrix that describes the position and rotation of the frame that corresponds to every link of the robot in a frame located at the torso of the robot. Figure 3.10 represents the "zero" pose of the robot and the rotations of the reference frame of every link for the Denavit-Hartenberg parameters expressed in Table 3.1.

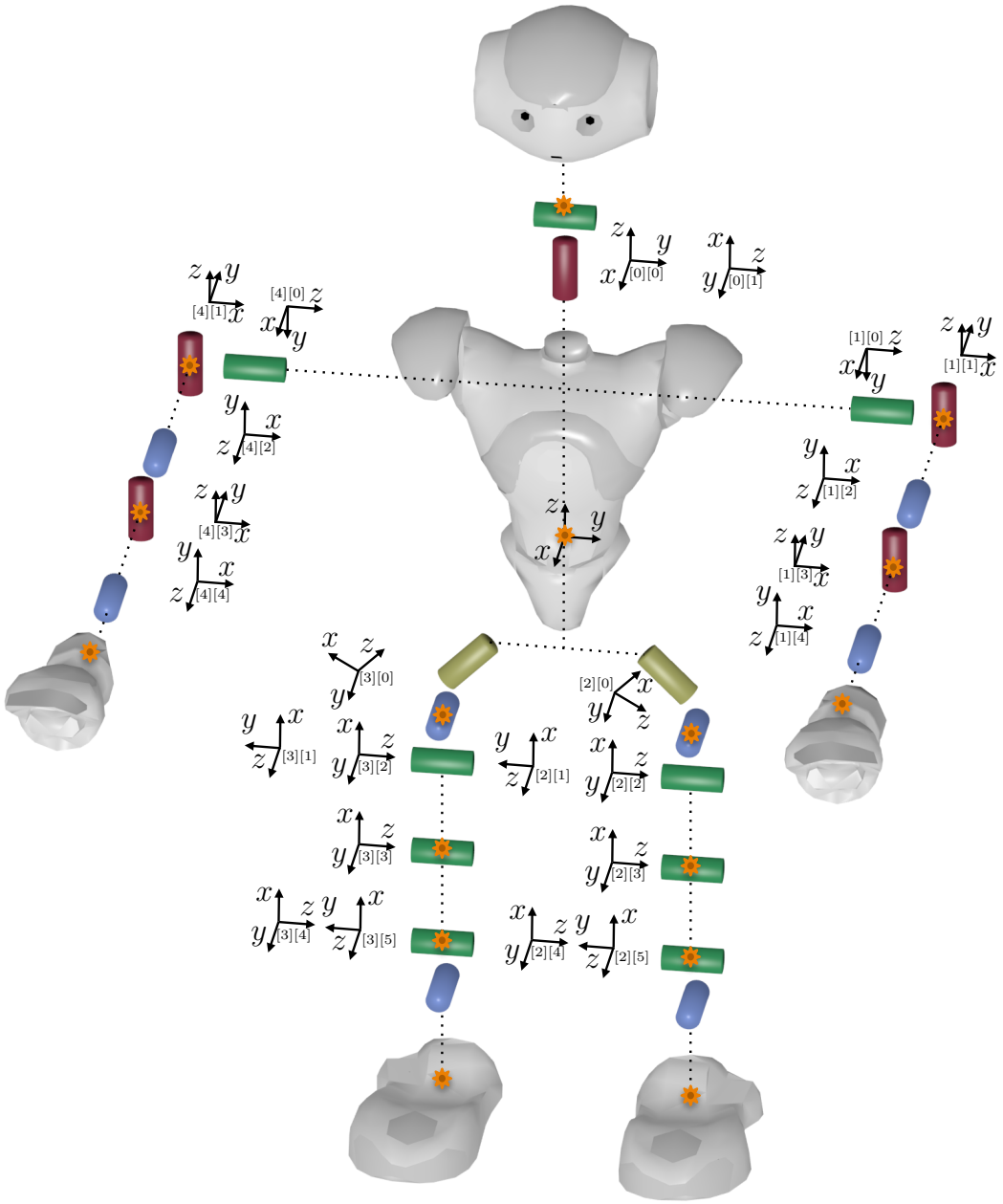


Figure 3.10: Denavit-Hartenberg joint definitions.

Joint Name	α	a	θ	d
HeadYaw	0	0	0	0
HeadPitch	$-\pi/2$	0	$-\pi/2$	0
LShoulderPitch	$-\pi/2$	0	0	0
LShoulderRoll	$\pi/2$	0	$\pi/2$	0
LElbowYaw	$\pi/2$	0	0	$UpperArmLength$
LElbowRoll	$-\pi/2$	0	0	0
LHipYawPitch	$-3\pi/4$	0	-$\pi/2$	0
LHipRoll	$-\pi/2$	0	$\pi/4$	0
LHipPitch	$\pi/2$	0	0	0
LKneePitch	$\pi/2$	$-ThighLength$	0	0
LAnklePitch	0	$-TibiaLength$	0	0
LAnkleRoll	$-\pi/2$	0	0	0
RHipYawPitch	$-\pi/4$	0	$-\pi/2$	0
RHipRoll	$-\pi/2$	0	-$\pi/4$	0
RHipPitch	$\pi/2$	0	0	0
RKneePitch	0	$-ThighLength$	0	0
RAnklePitch	0	$-TibiaLength$	0	0
RAnkleRoll	$-\pi/2$	0	0	0
RShoulderPitch	$-\pi/2$	0	0	0
RShoulderRoll	$\pi/2$	0	$\pi/2$	0
RElbowYaw	$\pi/2$	0	0	$UpperArmLength$
RElbowRoll	$-\pi/2$	0	0	0

Table 3.1: Denavit-Hartenberg parameters for Nao.

The modified Denavit-Hartenberg notation employed follows the convention established by the manufacturer of the robot, with the following meaning for the parameters:

- α: rotation from $\boldsymbol{Z}_{i-1}$ axis to $\boldsymbol{Z}_i$ axis around $\boldsymbol{X}_{i-1}$ axis.
- a: distance from $\boldsymbol{Z}_{i-1}$ axis to $\boldsymbol{Z}_i$ axis along $\boldsymbol{X}_{i-1}$ axis.
- θ: rotation from $\boldsymbol{X}_{i-1}$ axis to $\boldsymbol{X}_i$ axis around $\boldsymbol{Z}_i$ axis.
- d: distance from $\boldsymbol{X}_{i-1}$ axis to $\boldsymbol{X}_i$ axis along $\boldsymbol{Z}_i$ axis.

In this way, for a certain joint space, it is possible to obtain the homogeneous matrix for the reference system of every link. Since the position of the Center of Mass of every link is known, it is also possible to obtain the position of the global Center of Mass of the whole body. Finally, the pose of the sole frames using the reference frame located at the torso are also obtained. By transforming the reference frame from the torso to the sole frame of the support foot we achieve the transformation between the joint space and the com-torso-foot space. Let us see as an example the procedure to obtain the location of the CoM of the torso in the sole frame:

1. **Homogeneous matrix construction**. Once the rotation of the joint is added to the value of θ displayed in Table 3.1, the homogeneous matrix that represents the transformation from the previous reference system, $\boldsymbol{R}_{i-1}$, to the reference system of that link, $\boldsymbol{R}_i$, is constructed as follows:

$$\boldsymbol{T}_i = \begin{pmatrix} \cos\theta & -\sin\alpha & 0 & a \\ \cos\alpha\sin\theta & \cos\alpha\cos\theta & -\sin\alpha & -d\sin\alpha \\ \sin\alpha\sin\theta & \sin\alpha\cos\theta & \cos\alpha & d\cos\alpha \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.3}$$

2. **CoM position of a link in torso frame**. The position of the CoM of a link i in its local reference frame, $\boldsymbol{R}_i$, is given by $\boldsymbol{C}_i = (Cx_i, Cy_i, Cz_i)^t$. To obtain the position of the CoM of the link in the torso frame, we simply multiply the transformation matrix $\boldsymbol{T}_i$ of each link of the kinematic chain from the torso to the link i and finally multiply the resulting matrix by $\boldsymbol{C}_i$:

$$\boldsymbol{C}_{i_t} = \boldsymbol{T}_1\boldsymbol{T}_2...\boldsymbol{T}_i\boldsymbol{C}_i. \tag{3.4}$$

3. **Global CoM position in torso frame**. Provided the positions of the CoM of every link in the torso frame, the global CoM of the robot is simply the weighted average of the position of the CoM of every link. Let us recall the equation mentioned in 2.1:

$$\boldsymbol{C}_t = \frac{1}{M} \sum M_i \boldsymbol{C}_{i_t}. \tag{3.5}$$

4. **Sole frame pose in torso frame**. To obtain the pose of the reference frame of the sole of a foot in the torso frame we first move the distance $FootHeight$ along the normal to the sole plane at the end of the kinematic chain of a the corresponding leg. For example, for the left foot:

$$\begin{aligned} \boldsymbol{L}_T = \boldsymbol{T}_{LHipYawPich}\, \boldsymbol{L}_{LHipRoll}\, \boldsymbol{T}_{LHipPitch}\, \boldsymbol{T}_{LKneePitch} \\ \boldsymbol{T}_{LAnklePitch}\, \boldsymbol{T}_{LAnkleRoll} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -FootHeight \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{aligned} \tag{3.6}$$

5. **Torso frame in normalized sole frame**. The reference system obtained through the composition of transformation matrixes at the end of the kinematic chain of the leg is rotated with respect to the original. We are interested in obtaining a reference system located at the sole that has the same orientation than the torso when all joints are set at the zero position. For that, the system at the end of the kinematic chain of the leg needs to be pre-rotated around the Y and X axes. Then, the inverse of the matrix $\boldsymbol{S}_t$ is employed to reverse the transformation:

$$\boldsymbol{T}_L = \boldsymbol{T}_{RotX(\pi)} \boldsymbol{T}_{RotY(\frac{\pi}{2})} \boldsymbol{S}_t^{-1}. \tag{3.7}$$

Please note that the rotation transformation employed above take the form of homogenous matrixes:

$$\boldsymbol{T}_{RotX(\phi)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & C(\phi) & -S(\phi) & 0 \\ 0 & S(\phi) & C(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{3.8}$$

$$\boldsymbol{T}_{RotY(\phi)} = \begin{pmatrix} C(\phi) & 0 & S(\phi) & 0 \\ 0 & 1 & 0 & 0 \\ -S(\phi) & 0 & C(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{3.9}$$

$$\boldsymbol{T}_{RotZ(\phi)} = \begin{pmatrix} C(\phi) & -S(\phi) & 0 & 0 \\ S(\phi) & C(\phi) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{3.10}$$

where $C(\phi)$ and $S(\phi)$ are the abbreviations of $\cos(\phi)$ and $\sin(\phi)$, respectively.

6. **Transform CoM from torso frame to sole frame**. This is simply achieved by multiplying the matrix that describes the transformation from the sole frame to the torso frame ($\boldsymbol{C}_L$ for the left leg) and the position vector with the location of the CoM in the torso frame ($\boldsymbol{C}_t$):

$$\boldsymbol{C}_L = \boldsymbol{T}_L \boldsymbol{C}_t. \tag{3.11}$$

In the previous case, the goal of the process was to obtain the center of mass $\boldsymbol{C}_t$, which is defined by a vector. In other cases, the target of the procedure is to obtain the location and orientation of a certain part of the body. For instance it may be of interest to obtain the location and orientation of the sole of the swing foot in the reference frame located at the sole of the support foot. Assuming the left foot is the one that has the role of support foot, the pose of the normalized reference system of the sole of the right foot in the normalized frame of the left foot sole is given by:

$$\boldsymbol{R}_L = \boldsymbol{T}_L \boldsymbol{T}_R^{-1}. \tag{3.12}$$

To obtain the location Cartesian coordinates of a reference frame from the homogenous matrix that defines its pose, we simply take the first three rows of the last column of the homogeneous matrix:

$$\boldsymbol{P} = [H_{03}, H_{13}, H_{23}] \tag{3.13}$$

For the extraction of the roll-pitch-yaw angles, the process is slightly more complicated. The upper-left 3x3 matrix of the homogeneous matrix is selected and compared with the resulting matrix for a system rotated around the X, Y and Z axis. The roll, pitch and yaw angles can be then obtained by system identification.

3.6 Inverse Kinematics

The goal of the inverse kinematics approach is to obtain the angles for the joints that correspond to a given pose in the com-torso-foot space. Although general approaches exist that solve the inverse kinematics problem [32, 10, 84, 71], they involve the use of Jacobians, and are therefore computationally expensive. For a platform such as the Nao humanoid, a more efficient solution is required. The advantages of the use of analytical or mixed approaches instead of pure numerical ones has been pointed in [100]. These appraoches are available for other robots but, due to the special configuration of the coupled yaw joints of the hips of Nao, a customized solution had to be developed for this platform.

The first problem to address is the mismatch between the dimensions of the joint space and the com-torso-foot space. As we commented above, the representation of the position of the CoM requires 3 dimensions, the orientation of the torso needs 3 more and the full specification of the pose of the swing foot requires 6 additional dimensions. That makes a total account of 12 dimensions for the com-torso-foot space. Nonetheless, the number of joints in the kinematics chains of the legs is 11. This is not a problem for the forward kinematics but, in order to solve the inverse kinematics problem, one of the dimensions of the com-torso-foot space needs to be omitted. By removing the yaw angle specification of the swing foot we realized that an analytic solution could be obtained. Nevertheless, the yaw angle of the swing foot is important for locomotion, because it allows the robot to perform curved trajectories on the ground plane.

On the other hand, the yaw angle of the torso orientation is not critical neither for balance nor for locomotion trajectory performance. Therefore, we prefer to drop the control of this angle from the balance algorithms. Since there is a strong relationship between these two yaw angles for the typical movements of the robot, we will employ a P controller on the yaw angle of the torso to track the reference value for the yaw angle of the swing foot. Then, the swing foot yaw angle is omitted from the input to the next stages of the inverse kinematics approach.

Additionally, the use of the CoM involves taking into account the masses of the different parts of the body, which greatly complicates the analytical solution of the problem. To cope with this issue, we perform an additional stage in the

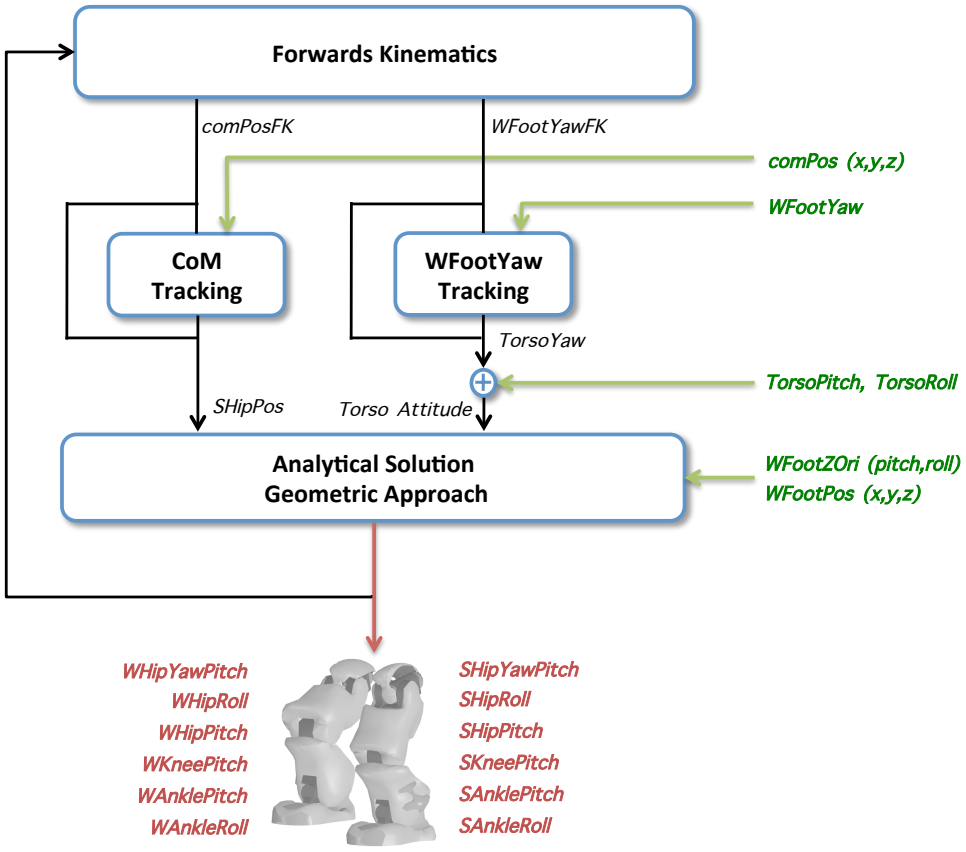


Figure 3.11: Inverse Kinematics Overview.

inverse kinematics procedure. In this stage, we move the hip extreme of the support leg of the robot to approach the target position of the CoM.

In the final stage, we employ a geometric approach to obtain analytically the joint angles that correspond to that position of the hip, and the given torso orientation and swing foot pose (without yaw angle). Since the solution of the previous stages are approximated, all three stages are included in a loop that can be iterated to improve accuracy. However, due to the constant updating in every locomotion control cycle, we found that no iterations were needed to obtain satisfactory accuracy.

The schema of the inverse kinematics approach is displayed on Figure 3.11. Since most of the operations are the same for any leg, we will do the calculations for a generic support leg (S prefix) and swing leg (W prefix).

3.6.1 Swing foot yaw angle tracking

The first stage of the inverse kinematics process is the tracking of the swing foot yaw angle (WFootYaw). This goal is simply performed by a P-controller:

$$\text{TorsoYaw}_i = \text{TorsoYaw}_{i-1} + 0.5 \cdot (\text{WFootYaw}_i - \text{WFootYawFK}_i). \tag{3.14}$$

The drawback of this simple procedure is that it is only suitable for standard poses of the robot. In case the sole of the swing foot is not parallel to the sole of the support leg or the torso deviates from the normal to the support sole, the performance of this approach decreases. Hence it is advisable in such situations to avoid employing yaw angles for the swing foot, or if necessary, to take additional precautions.

3.6.2 CoM tracking

Since the position of the hip extreme of the support leg ($\boldsymbol{SHipPos}$) is linked to the trunk of the robot, we will focus on the calculation of that position. The three dimensional position of the CoM of the robot ($\boldsymbol{CoM^{ref}}$) is calculated by considering separately the position of the CoM of the support leg ($\boldsymbol{CoM^{l}}$) and the position of the CoM of the rest of the body ($\boldsymbol{CoM^{b}}$). Similarly, the mass of the support leg (M^l) and the mass of the rest of the body (M^b) are considered separately, being M the whole mass of the robot.

In this way, we will calculate the increment of the position of the hip extreme of the support leg ($\Delta\boldsymbol{SHipPos}$) that moves the CoM towards its target position. Even though $\Delta\boldsymbol{SHipPos}$ is applied directly to $\boldsymbol{CoM^{b}}$, we assume that it only affects to $\boldsymbol{CoM^{l}}$ in a fraction that it is neglected. Expressed the positions as 3×1 vectors with its Cartesian coordinates, the location of the CoM after the movement of the support hip is calculated as follows:

$$\boldsymbol{CoM^{ref}} \cdot M = (\boldsymbol{CoM^{b}} + \Delta\boldsymbol{SHipPos}) \cdot M^b + \boldsymbol{CoM^{l}} \cdot M^l, \tag{3.15}$$

which permits to obtain the position of the support hip ($\boldsymbol{SHipPos}$) by an iterative process that adds $\Delta\boldsymbol{SHipPos}$ to the previous position of the hip. Since the inverse kinematics problem must be solved in every control cycle of the platform, only one iteration is required to achieve satisfactory accuracy in the control of the CoM. Figure 3.12 represents the tracking process in the i-th iteration.

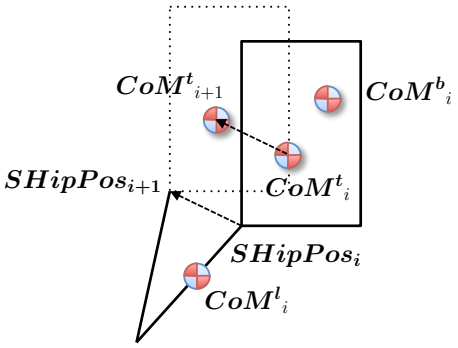


Figure 3.12: The position of the hip is modified to track the target location of the CoM.

3.6.3 Analytical Geometric Solution

Once the yaw angle of the torso and the position of the hip extreme of the support leg are calculated, these results are combined with the remaining input data to solve the inverse kinematics problem. The last stage of the algorithm is analytical, and its inputs are:

- 3x1 position vector of the hip extreme of the support leg, $\boldsymbol{SHipPos}$.
- 3x3 rotation matrix of the torso $\boldsymbol{O_{torso}}$.
- 3x1 position vector of the swing foot, $\boldsymbol{WFootPos}$.
- 3x1 coordinates of the Z axis unitary vector of the swing foot sole frame, $\boldsymbol{WFootZOri}$.

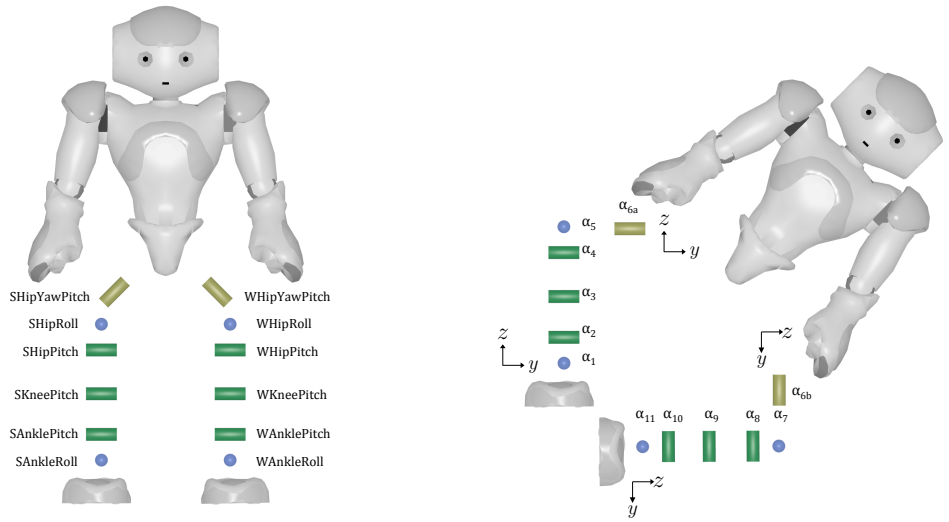
The rotation values of the joints are represented by the α symbol and a subindex corresponding to the position occupied in the kinematic chain that starts in the support foot and ends in the swing foot. To simplify the calculations, the zero position of the roll angles of the hip joints are shifted by $\pi/4$ to set them parallel to the pitch joints, as displayed in Figure 3.13. The sign of the $\pi/4$ offset of the α joints depends on the leg that is employed as support, but will be always positive for the roll joint of the right side of the hip and negative for the roll joint of the left side. The equivalences between α codes and the joints are detailed in Table 3.2.

According to the schema presented in 3.13b, the orientation of the frame located in the sole of the swing foot, can be obtained through the following sequence of rotations:

$$\begin{aligned}\boldsymbol{O_{11}} =& \boldsymbol{R}_x(-\alpha_1)\boldsymbol{R}_y(-\alpha_2)\boldsymbol{R}_y(-\alpha_3)\boldsymbol{R}_y(-\alpha_4)\boldsymbol{R}_x(-\alpha_5)\boldsymbol{R}_y(-\alpha_6) \\ & \boldsymbol{R}_x(\pi/2)\boldsymbol{R}_y(\alpha_6)\boldsymbol{R}_x(\alpha_7)\boldsymbol{R}_y(\alpha_8)\boldsymbol{R}_y(\alpha_9)\boldsymbol{R}_y(\alpha_{10})\boldsymbol{R}_x(\alpha_{11}).\end{aligned} \tag{3.16}$$

Code	Joint
α_1	$SAnkleRoll$
α_2	$SAnklePitch$
α_3	$SKneePitch$
α_4	$SHipPitch$
α_5	$SHipRoll \mp \pi/4$
α_{6a}	$SHipYawPitch$
α_{6b}	$WHipYawPitch$
α_7	$WHipRoll \pm \pi/4$
α_8	$WHipPitch$
α_9	$WKneePitch$
α_{10}	$WAnklePitch$
α_{11}	$WAnkleRoll$

Table 3.2: Joint index codes.



(a) Original configuration. (b) Shifted configuration.

Figure 3.13: Hip roll joints are shifted to facilitate analytic calculations.

It is interesting to notice that the angles of the α values corresponding to the joints of the support leg are preceded by the negative sign. This is due to the sign convention, which considers that the rotation of a limb around and axis is positive if the connected limb is closer to the torso in the kinematics chain. In the remaining of the section we will explain the algorithms employed to obtain analytically the positions of the joints.

3.6.3.1 SAnkleRoll, SAnklePitch and SKneePitch

The first values to be obtained correspond to the joint and the knee of the support leg. These joints are only dependent on the position of the extreme of the leg that links it with the trunk, $\boldsymbol{SHipPos}$. To start the calculation, we obtain the distance $\boldsymbol{SAnkleSHip}$ from the ankle of the support foot to the opposite extreme of the leg, $\boldsymbol{SHipPos}$. This is achieved by simply subtracting the height of the foot, $\boldsymbol{FH}(0, 0, FootHeight)$, to the position of the hip.

$$\boldsymbol{SAnklePos} = \boldsymbol{SHipPos} - \boldsymbol{FH} \tag{3.17}$$

The problem of placing the hip at the target place can be divided in two parts. The first one aims at heading the extreme of the leg in the right direction, that is, to make a straight leg go through $\boldsymbol{SHipPos}$. The goal of the second part is to reduce the length of the leg to the amount required. To begin with the first problem let us calculate the ankle values for a telescopic leg that will place the hip at the target position:

$$\alpha_{1A} = \arctan\left(\frac{SAnkleSHip_y}{SAnkleSHip_z}\right) \tag{3.18}$$

$$\alpha_{2A} = \arcsin\left(\frac{-SAnkleSHip_x}{SL}\right), \tag{3.19}$$

where SL stands for the module of the distance between the ankle and the hip extremes extremes of the leg:

$$SL = \|\boldsymbol{SAnkleSHip}\|. \tag{3.20}$$

Next, we will find the angle that is necessary to bend the knee the set the distance between the ankle and the hip to SL, as depicted in Figure 3.14. Furthermore, we can obtain the angle of the ankle that counteracts the rotation of the knee, allowing the leg to behave as a telescopic system. Since the size of the thigh (THL) and the tibia (TBL) are known, we can apply the cosine theorem to obtain both angles:

$$\alpha_{2B} = \arccos\left(\frac{TBL^2 + SL^2 - THL^2}{2 \cdot TBL \cdot SL}\right) \tag{3.21}$$

$$\alpha_{3B} = \arccos\left(\frac{TBL^2 + THL^2 - SL^2}{2 \cdot TBL \cdot THL}\right) \tag{3.22}$$

The angles for the first three joints are obtained by combining the results of these two solutions:

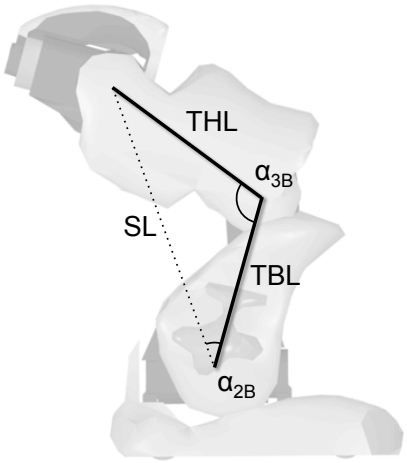


Figure 3.14: Hip-knee-ankle plane of the support leg.

$$\alpha_1 = \alpha_{1A} \tag{3.23}$$
$$\alpha_2 = \alpha_{2A} - \alpha_{2B} \tag{3.24}$$
$$\alpha_3 = \pi - \alpha_{3B} \tag{3.25}$$

The negative sign for α_{2B} and the subtraction of π from α_3 are due to the sign convention employed.

3.6.3.2 SHipPitch, SHipRoll and HipYawPitch

To obtain the joints located at the support side of the hip we will first rotate the original frame of the support sole according to the values obtained for α_1, α_2 and α_3. The orientation of the frame located right after the rotation of the knee is given by:

$$\boldsymbol{O_3} = \boldsymbol{R}_x(-\alpha_1)\boldsymbol{R}_y(-\alpha_2)\boldsymbol{R}_y(-\alpha_3). \tag{3.26}$$

In order to find the angles for the three joints of the support side of the hip, we will exploit the information of the orientation of the torso, which is known:

$$\boldsymbol{O_3}\boldsymbol{R}_y(-\alpha_4)\boldsymbol{R}_x(-\alpha_5)\boldsymbol{R}_y(-\alpha_6)\boldsymbol{R}_x(-\frac{\pi}{4}) = \boldsymbol{O_{torso}} \tag{3.27}$$

Operating at both sides of $\boldsymbol{O_{torso}}$, we can obtain the numerical solution for the rotation matrix of the three joints of the hip at the support side:

$$\boldsymbol{R_{hs}} = \boldsymbol{O_3}^t \boldsymbol{O_{torso}} \boldsymbol{R}_x(\frac{\pi}{4}) \tag{3.28}$$

At the same time, the algebraic solution of $\boldsymbol{R_{hs}}$ can be obtained by performing the rotations of the hip joints. For the sake of simplicity, we will employ S_i and C_i to represent the sine and the cosine of the α_i angle.

$$\begin{aligned} \boldsymbol{R_{hs}} &= \begin{pmatrix} C_4 & 0 & -S_4 \\ 0 & 1 & 0 \\ S_4 & 0 & C_4 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & C_5 & S_5 \\ 0 & -S_5 & C_5 \end{pmatrix} \begin{pmatrix} C_6 & 0 & -S_6 \\ 0 & 1 & 0 \\ S_6 & 0 & C_6 \end{pmatrix} \\ &= \begin{pmatrix} C_4C_6 - S_4C_5S_6 & S_4S_5 & -C_4S_6 - S_4C_5C_6 \\ S_5S_6 & C_5 & S_5C_6 \\ S_4C_6 + S_4C_5S_6 & -C_4S_5 & -S_4S_6 + C_4C_5C_6 \end{pmatrix} \end{aligned} \tag{3.29}$$

Then, the values for the joints α_4, α_5 and α_6 can be obtained by identifying both expressions of R_{sh}:

$$\alpha_4 = \arctan(-R_{hs_{12}}/R_{hs_{32}}) \tag{3.30}$$

$$\alpha_5 = \arcsin(-R_{hs_{32}}/\cos(\alpha_4)) \tag{3.31}$$

$$\alpha_6 = \arctan\left(\frac{\sin(\alpha_4) * \frac{R_{hs_{11}}}{R_{hs_{31}}} - \cos(\alpha_4)}{-\sin(\alpha_4) * \cos(\alpha_5) - \frac{R_{hs_{11}}}{R_{hs_{31}}} * \cos(\alpha_4) * \cos(\alpha_5)}\right) \tag{3.32}$$

3.6.3.3 WHipRoll

Since the hipYawPitch joints of both legs are coupled, it is possible to calculate the orientation of the frame after the WHipYawPitch joint:

$$\boldsymbol{O_{6w}} = \boldsymbol{O_3}\boldsymbol{R}_y(-\alpha_4)\boldsymbol{R}_x(-\alpha_5)\boldsymbol{R}_y(-\alpha_6)\boldsymbol{R}_x(-\pi/2)\boldsymbol{R}_y(\alpha_6). \tag{3.33}$$

At this point our goal is to calculate the roll angle for the hip of the swing leg that will place the sole of the swing foot at the target position. This joint defines the hip-knee-ankle plane of the swing leg. But first we need to find the direction of the vector that links the ankle and the hip of the swing leg.

Given the pose of the sole of the swing foot, we can calculate the location of the ankle of this foot:

$$\boldsymbol{WAnklePos} = \boldsymbol{WFootPos} + \boldsymbol{WFootZOri} \cdot FootHeight \tag{3.34}$$

In the same way, provided the position of the hip at the extreme of the support hip ($\boldsymbol{SHipPos}$), the orientation of the torso ($\boldsymbol{O_{torso}}$) and the distance vector between both sides of the hip, $\boldsymbol{HD}(0, HipDistance, 0)'$, we can calculate the position of the point of the hip that is linked to the swing leg:

$$\boldsymbol{WHipPos} = \boldsymbol{SHipPos} + \boldsymbol{O_{torso}HD} \tag{3.35}$$

The vector that links the ankle and the hip side of the support leg is therefore:

$$\boldsymbol{WAnkleWHip} = \boldsymbol{WHipPos} - \boldsymbol{WAnklePos}. \tag{3.36}$$

Since $\boldsymbol{WAnkleWHip}$ belongs to the plane hip-knee-ankle of the swing leg, we can find the roll joint of the hip by forcing the Y axis of the frame $\boldsymbol{O_7}$ to be perpendicular to $\boldsymbol{WAnkleWHip}$. To achieve this, we first use a symbolic angle α_7 to rotate the system $\boldsymbol{O_{6w}}$:

$$\begin{aligned} \boldsymbol{O_7} = \boldsymbol{O_{6w}}\boldsymbol{R}_x(\alpha_7) &= \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & C_7 & -S_7 \\ 0 & S_7 & C_7 \end{pmatrix} \\ &= \begin{pmatrix} t_{11} & t_{12}C_7 + t_{13}S_7 & -t_{12}S_7 + t_{13}C_7 \\ t_{21} & t_{22}C_7 + t_{23}S_7 & -t_{22}S_7 + t_{23}C_7 \\ t_{31} & t_{32}C_7 + t_{33}S_7 & -t_{32}S_7 + t_{33}C_7 \end{pmatrix} \end{aligned} \tag{3.37}$$

And secondly, we force the scalar product of the Y axis of O_7 and $\boldsymbol{WAnkleWPelvis}$ to be zero:

$$\left(t_{12}C_7 + t_{13}S_7, t_{22}C_7 + t_{23}S_7, t_{32}C_7 + t_{33}S_7\right) \cdot \boldsymbol{WAnkleWPelvis} = 0 \tag{3.38}$$

Solving for α_7, we get:

$$\alpha_7 = \arctan\left(\frac{\boldsymbol{O_{6w}}[:,2] \cdot \boldsymbol{WAnkleWPelvis}}{\boldsymbol{O_{6w}}[:,3] \cdot \boldsymbol{WAnkleWPelvis}}\right) \tag{3.39}$$

3.6.3.4 WHipPitch and WKneePitch

The next goal is to find the values for the pitch joints of the hip and the knee of the swing leg that place the ankle at the target position. The strategy will be similar to the one used for the support foot. First we will assume a telescopic leg and find the angle of the hip pitch to reach the target position with the ankle. Then, we will calculate the required rotation in the hip and in the knee to reduce the dimension of the leg with the cosine theorem.

In the case of a telescopic leg, to calculate the rotation of the hip, we will look for the angle that will place the x axis of the frame perpendicular to the ***WAnkleWPelvis*** vector.

$$\begin{aligned} \boldsymbol{O_{8A}} = \boldsymbol{O_7 R}_y(\alpha_8) &= \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{pmatrix} \begin{pmatrix} C_{8A} & 0 & S_{8A} \\ 0 & 1 & 0 \\ -S_{8A} & 0 & C_{8A} \end{pmatrix} \\ &= \begin{pmatrix} t_{11}C_{8A} - t_{13}S_{8A} & t_{12} & t_{11}S_{8A} + t_{13}C_{8A} \\ t_{21}C_{8A} - t_{23}S_{8A} & t_{22} & t_{21}S_{8A} + t_{23}C_{8A} \\ t_{31}C_{8A} - t_{33}S_{8A} & t_{32} & t_{31}S_{8A} + t_{33}C_{8A} \end{pmatrix} \end{aligned} \tag{3.40}$$

$$(t_{11}C_{8A} - t_{13}S_{8A}, t_{21}C_{8A} - t_{23}S_{8A}, t_{31}C_{8A} - t_{33}S_{8A}) \cdot \boldsymbol{WAnkleWPelvis} = 0 \tag{3.41}$$

$$\alpha_{8A} = \arctan\left(\frac{\boldsymbol{O_7}[:,1] \cdot \boldsymbol{WAnkleWPelvis}}{\boldsymbol{O_7}[:,3] \cdot \boldsymbol{WAnkleWPelvis}}\right) \tag{3.42}$$

And now, to reduce the length of the leg, we employ the cosines theorem again:

$$\alpha_{8B} = \arccos\left(\frac{TBL^2 + wL^2 - THL^2}{2 \cdot THL * wL}\right) \tag{3.43}$$

The pitch angle of the hip side of the swing leg, is therefore:

$$\alpha_8 = \alpha_{8A} - \alpha_{8B} \tag{3.44}$$

Similarly, the rotation for the knee is:

$$\alpha_9 = \pi - \arccos\left(\frac{TBL^2 + THL^2 - wL^2}{2 * TBL * THL}\right) \tag{3.45}$$

3.6.3.5 WAnklePitch

At this point, the ankle of the swing foot has reached its target position, and the last task is to rotate the ankle joints to place the sole of the swing foot with the required orientation. The orientation of the sole plane of the swing foot is given by the vector of the Z axis of the sole frame of the swing foot. This vector is normal to the plane of the sole. Therefore, we will first use the pitch joint of

the ankle of the swing foot to place the X axis of the frame perpendicular to the required Z axis:

$$\begin{aligned}\boldsymbol{O_{10}} = \boldsymbol{O_9}\boldsymbol{R}_y(\alpha_{10}) &= \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{pmatrix} \begin{pmatrix} C_{10} & 0 & S_{10} \\ 0 & 1 & 0 \\ -S_{10} & 0 & C_{10} \end{pmatrix} \\ &= \begin{pmatrix} t_{11}C_{10} - t_{13}S_{10} & t_{12} & t_{11}S_{10} + t_{13}C_{10} \\ t_{21}C_{10} - t_{23}S_{10} & t_{22} & t_{21}S_{10} + t_{23}C_{10} \\ t_{31}C_{10} - t_{33}S_{10} & t_{32} & t_{31}S_{10} + t_{33}C_{10} \end{pmatrix}\end{aligned} \tag{3.46}$$

$$\alpha_{10} = \arctan\left(\frac{\boldsymbol{O_9}[:,1]) \cdot \boldsymbol{WFootZOri}}{\boldsymbol{O_9}[:,3]) \cdot \boldsymbol{WFootZOri}}\right) \tag{3.47}$$

3.6.3.6 WAnklePitch

Finally, the roll angle of the ankle will rotate the frame around the X axis to set the Y axis perpendicular to the Z axis.

$$\begin{aligned}\boldsymbol{R}_{11} = \boldsymbol{O_{10}}\boldsymbol{R}_x(\alpha_{10}) &= \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & C_{11} & -S_{11} \\ 0 & S_{11} & C_{11} \end{pmatrix} \\ &= \begin{pmatrix} t_{11} & t_{12}C_{11} + t_{13}S_{11} & -t_{12}S_{11} + t_{13}C_{11} \\ t_{21} & t_{22}C_{11} + t_{23}S_{11} & -t_{22}S_{11} + t_{23}C_{11} \\ t_{31} & t_{32}C_{11} + t_{33}S_{11} & -t_{32}S_{11} + t_{33}C_{11} \end{pmatrix}\end{aligned} \tag{3.48}$$

$$\alpha_{11} = \arctan\left(\frac{\boldsymbol{O_{10}}[:,2] \cdot \boldsymbol{WFootZOri}}{\boldsymbol{O_{10}}[:,3] \cdot \boldsymbol{WFootZOri}}\right) \tag{3.49}$$

3.7 Discussion

Along the previous sections we have reviewed the special features of the Nao platform and we have analyzed the performance of its sensors and actuators. Moreover, we have showed the calculations required to solve the forward kinematics and we have proposed an approach specially tailored to solve the inverse kinematics of the Nao robot. This approach combines an analytical geometric solution with proportional-like control for the CoM and the yaw angle of the swing foot.

The conclusions of the analysis of the Nao platform are:

- The FSR sensors are not appropriate to estimate the position of the CoP.

- The FSR sensors are suitable to detect ground contact.
- The 3-axis accelerometer unit is not appropriate to measure external forces produced by low level impacts.
- The 3-axis accelerometer can be used to estimate the direction of gravity and therefore to obtain an approximate estimation of the orientation of the torso.
- The 2-axis gyrometer provides accurate measures of the rotational velocity of the torso. These measures can be integrated to obtain an estimate of the orientation of the torso but the error of the readings is cumulative.
- Information provided by accelerometer, gyrometer and FSR sensors can be fused in a Kalman filter to obtain a satisfactory estimation of the pitch and roll angles of the torso.
- The sensory-motor lag of the platform is equivalent to 4 control cycles.
- If the sign of the acceleration of the torso changes abruptly, the play accumulated along the joints of the legs can be up to 10° for the sagittal plane and up to 4° for the frontal plane.
- The number of DoF of the legs does not allow a full specification of poses in the com-torso-foot space. Therefore, the yaw rotation of the torso is not considered for the inverse kinematics solution.

Chapter 4

Walking Pattern Generation

4.1 Introduction

The goal of a locomotion system is to generate the sequence of poses for the robot that will permit it to attain a certain target location or velocity. Through the use of an inverse kinematics module it is possible to specify these poses in the convenient Cartesian space. In this chapter we will start by describing the different modules required for walking pattern generation. Next, we review the procedure employed to derive the location of the footsteps from the motion type requested. Subsequently, the algorithms employed to generate the trajectories for the feet and the CoM of the robot will be detailed and, finally, we will examine the walking pattern generated for straight and curved motions.

4.2 Locomotion Architecture

The architecture of the locomotion system employed for this work is depicted in Figure 4.1. According to this schema, the system receives a three-dimensional (sagittal, lateral and rotational) target velocity for the robot in the ground plane. The output of the walking pattern generator is an array of positions for the joints actuators. The commanded joint positions are tracked by the low-level PID algorithms implemented in the microcontrollers provided by the manufacturer. In this chapter we deal with the algorithms used to generate the patterns that will allow the robot to perform the gait. In order to attain a well-balanced locomotion, however, it is fundamental to take advantage of the information provided by the exteroceptive sensors, but this part is left for the next chapter.

As mentioned in the Forward and Inverse Kinematics sections, we combine the use of two different spaces to manipulate the robot. The first one is the joint space, which is required to communicate with the middleware provided by the

robot manufacturer to command it. This will be the space employed to control the joints of the neck and the arms but, since we are not using these parts of the body for any task related to the gait, the generation of the joint positions for that parts is out of the scope of this work.

In the joint space, the pose of the robot is represented as a set of rotational positions of all its joints. On the other hand, the com-torso-foot space is useful to model the dynamics of the system and to simplify the design of trajectories for the limbs. This space represents the pose of the robot by the position of the CoM the orientation of the torso and the pose of the swing foot. The orientation of the torso, however, refers only to its roll and pitch angle, to reduce the number of dimensions of the space and match the DoF of the legs.

In this way, CoM location, torso orientation and swing foot pose are related to a reference frame located at the sole of the support foot. The origin of coordinates of the reference frame for our system is placed at the ankle projection on the sole of the support foot. The X axis points forward, the Y axis leftwards and the Z axis upwards. The roll angles are rotations around the X axis of this frame and the pitch angles are rotations around the Y axis. Since the support foot role alternates between both feet, when the robot is walking, the system of coordinates jumps from one foot to the other in every step.

The first stage to obtain the pose of the robot in the com-torso-foot space is the generation of the target trajectory of the robot. In this regard, we will employ a flat horizontal ground assumption. According to the sagittal, lateral and rotational velocity that are required, the *Footstep Planner* generates a sequence with the following target positions for the footsteps of the robot.

Once the footsteps are ready, the feet of the robot execute a swing trajectory, alternatively, to reach the new positions of the footsteps route. The trajectory that a foot follows in the air is calculated by the *Swing Foot Pattern Generator*. The output of this module is a sequence of Cartesian positions and a rotation matrix of the nonsupporting foot in the support foot frame.

Additionally, to prevent the robot from falling, it is necessary to ensure certain stability conditions. In our case, we will employ the inverted pendulum model and the ZMP stability criterion. The dynamic balance condition resides in keeping the ZMP within the limits of the support polygon (single-support stage) or the convex hull (double-support stage). Since the feet move to track the footsteps sequence defined, the shape and position of the convex hull will vary. Hence we will define trajectories that move the ZMP to keep it within the limits of the convex hull. These trajectories are generated in the *ZMP Trajectory Planner module*.

With the trajectories obtained by the *ZMP Trajectory Planner* and the estimation of the current position of the CoM, it is possible to employ an optimum control system called Model Predictive Controller to generate the future positions

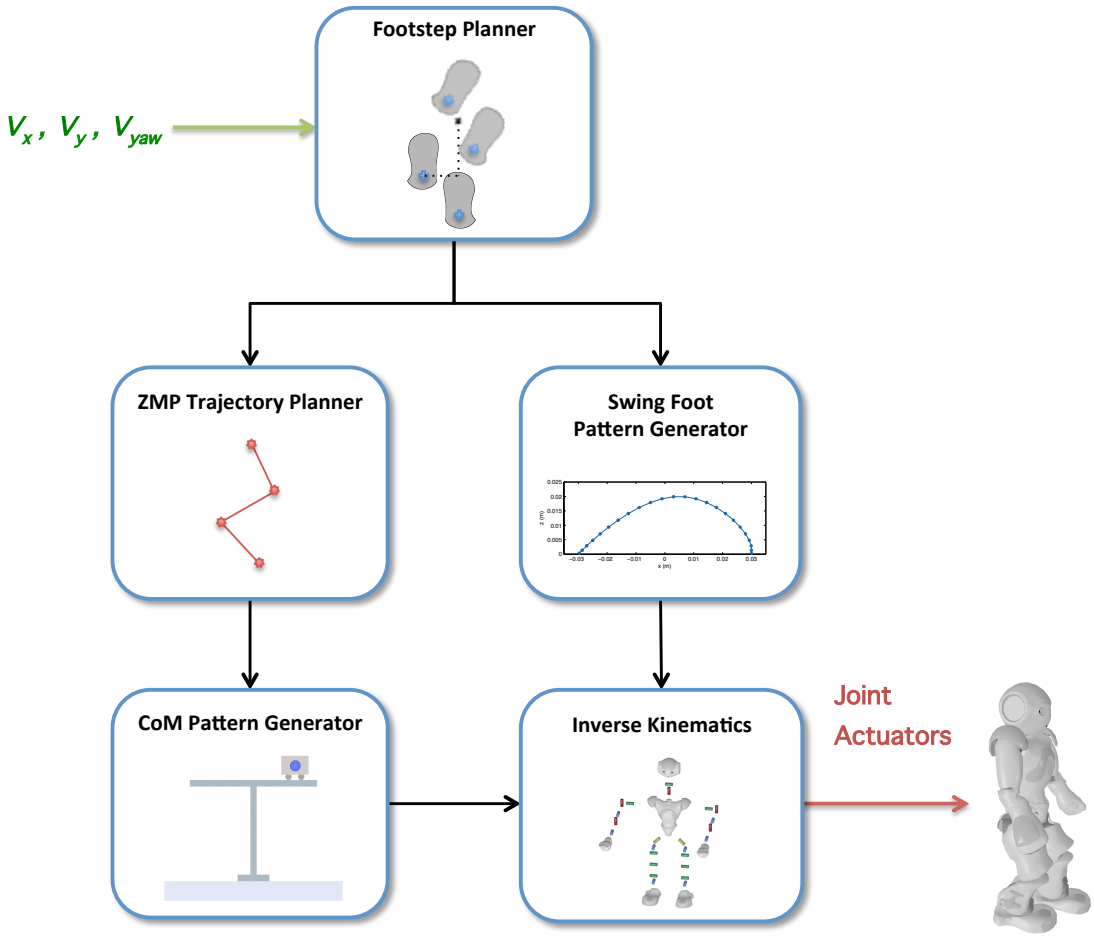


Figure 4.1: Architecture of the locomotion system.

of the CoM that minimize the deviations of the ZMP from the reference trajectory, as will be detailed in Section 4.5. This task is performed in the *CoM Pattern Generator*. The resulting CoM trajectory is delivered to the *Inverse Kinematics* module that, together with the swinging-foot pattern, will define the pose of the robot in the next control cycle.

The last component of the com-torso-foot space is the orientation of the torso. Although it has an impact in the balance of the body, it is not as determinant as the trajectory of the CoM and the feet. For this reason, in this work we will not consider the effect of the dynamics of the torso in the balance of the robot, and it will be set perpendicular to the sole plane in what follows.

4.3 Footstep Planing

Finding the optimum location for a step is a complex problem that has not yet been solved. Hence, many developments employ empirical approaches to generate the footstep locations. In this work we use a combination of empirical and analytic results to find the solution to the problem.

The optimal footstep depend on several factors. The first one is related to the linear and rotational inertia of the robot. If abrupt changes in the sense of the motion happen, the balance control algorithms are stressed. Although these algorithms provide flexibility in the selection of trajectories for the robot, taking the balance of the robot to its limit cause the platform to be less robust against external disturbances. A smooth design of the trajectories involves, on the other hand, a more robust performance. Our strategy in this regard is simple: we limit the maximum variation in each component (sagittal, lateral and yaw rotation) of the size of consecutive strides.

The second factor to be considered in the planning of the footsteps is the range of the footstep. The maximum size of a step depends of the configuration of the kinematic chain of the robot. The size of the segments that compose the leg, the rotational limits of the joints and even the location of the hip in the instant of support foot exchange have an impact in the shape of the potential destinations for the swing foot. The dynamic and high order nature of the function that defines this area hinders the analytic representation of the boundaries of the range of the kinematic chain. For this reason, we employ a simplified model of the kinematics with a conservative definition of the footstep range. This model involves three assumptions:

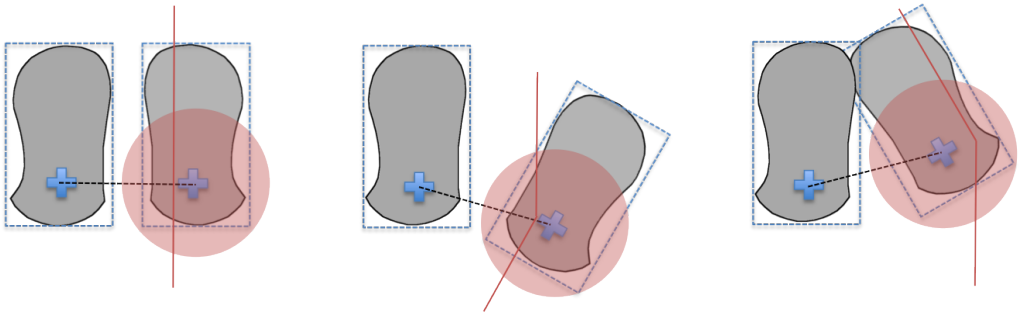
- The hip extreme of the support leg is located in the vertical line that goes through the ankle of the support foot.

- The ground projection of the sagittal plane of the torso bisects the angle between the longitudinal axis of booth feet.

- The area that can be reached by the swing leg can be simplified by a circle of a predefined radium whose center is located in the vertical projection on the ground of the hip extreme of the swing leg.

Figure 4.2 illustrates the potential footstep areas obtained by this model for different motion types. In all three cases, the left foot is employed to support the robot and the red area illustrates the potential locations for the right foot. The blue cross of the left foot indicates the position of the projection of the ankle of the left foot on the ground. The blue cross on the right, on the other hand, indicates the projection of the hip extreme of the right leg. The dotted line represents the projection of the hip. The potential locations for the step that could be reached by the leg are represented by a reddish circle. However, this areas are also limited by the last factor, which is the collision between the feet.

If a collision happens, one of the internal sides of the feet will always be involved, then we can distinguish two cases. If the turning motion describes a concave trajectory for the robot, then, one of the the top-inner corners of the soles will also be involved. Otherwise, it will be one of the back-inner corners the one involved in the collision. To obtain the collision limits, we model the soles of the feet as a rectangle. Then we rotate one of the feet to the requested angle and match the corners of the sole that could be involved in a collision for that kind of motion. The virtual location of the ankle of the swing foot will be the initial point of two half-lines that define the collision boundaries. Each of the half-lines runs parallel to the longitudinal axis of one of the feet. The half-lines divide the circular region calculated before in two parts, as shown in Figure 4.2.



(a) Zero-yaw motion. (b) Negative yaw motion. (c) Positive yaw motion.

Figure 4.2: The footstep area is limited by the range of the kinematic chain and the collision boundaries.

The external region is the area that we will allow for the location of the ankle of the swing foot. In case our target position is beyond this area, the closest point to it will be found and used instead. If that is the case, the calculated yaw angle for the footstep is preserved and the sagittal and lateral distances are adapted.

4.4 Foot Trajectory Design

Once computed the target footstep location, let us focus on the trajectory of the stride that will take the swing foot to its destination. The trajectories of the swing foot can induce a tipping moment on the robot due to the reaction forces generated by the collisions against the ground or to the inertial forces that appear when the foot is accelerated. Moreover, high inertial forces also prevent the joint actuators from effectively tracking the commanded positions.

In this work, the trajectories of the swing foot are performed in open-loop and the soles of the feet are kept parallel to each other. For that reason, the impacts of the swing foot against the ground are likely to occur at the point which heads the foot when it is approaching to the landing instant. For instance, if the robot walks forward, the stronger impacts against the ground will probably take place at the toe of the swing foot.

The sense of the reaction force, F_r, caused by the impact between the swing foot and the ground, depends on the direction of the velocity of the swing foot. The situation is described in Figure 4.3. The velocity of the foot decomposes into two components: one that is parallel to the sole plane, v_s, and another one that is normal to that plane, v_z. It can be assumed that the components of the reaction force in the direction of the sole plane, F_s, and in the normal direction to the sole plane, F_z, depend on the components of swing foot velocity v_s and v_z. That is, $F_s = F(v_s)$ and $F_z = F(v_z)$. The torque τ_s induced by the reaction force F_s will accelerate the tipping over. On the other hand, the torque τ_z generated by F_z contributes to recovering the upright pose.

Our aim is therefore to prevent collisions with a high component of F_s. Good strategies for this purpose are to raise the height of the stride and to reach the destination position in the horizontal plane before the end of the stride. However, the time available to perform the stride is conditioned by the target speed of the robot. The increase in the height of the stride and the reduction of the moment in the horizontal plane is done at the cost of increasing the accelerations applied to the swing foot. Hence, a trade-off between the height of the stride and the acceleration of the swing foot must be found. Figure 4.4 shows, in the sagittal plane, the stride design that we have selected for standard forward walking. Only 90% of the stride duration is employed for the movement in the transversal plane.

The implementation of the trajectory of the swing foot is based on cubic Bézier

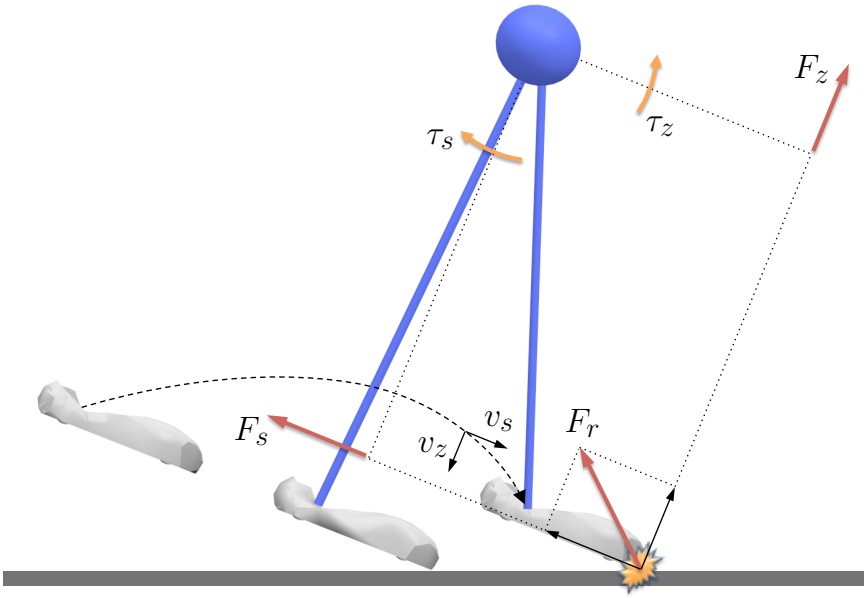


Figure 4.3: Ground impact during unbalanced step.

splines to generate the trajectory of the six dimensions that define the pose of the swing foot. Cubic splines are the lowest order polynomial endowed with inflection points, and the Bézier form of the polynomials provides a convenient design tool. This approach allows online modification of the landing position and provides flexibility to generate complex trajectories.

Cubic Bézier curves are defined by four control points P_0, P_1, P_2 and P_3 in the interval $0 <= t <= 1$ as in (4.1). The points P_0 and P_3 define the starting and the ending points of the trajectory, while the points P_1 and P_2 provide information to calculate the slope of the curve at the points P_0 and P_3 respectively.

$$b(t) = \sum_{i=0}^{3} \binom{3}{i} t^i (1-i)^{3-1} P_i \tag{4.1}$$

The design of complex trajectories can be achieved by the combination of several Bézier curves. In order to preserve the continuity and the differentiability, we have to apply the constraints (4.2) and (4.3) to the points of adjacent Bézier curves. The super index j of the control points indicate the sequential index of the Bézier curve, and T^j denote the duration of the Bézier curve j.

$$P_3{}^j = P_0{}^{j+1} \tag{4.2}$$

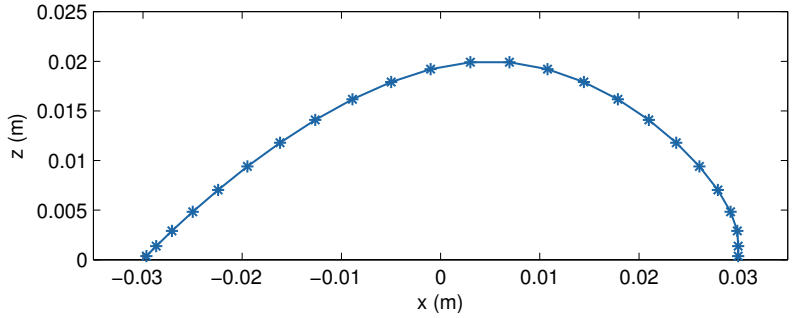


Figure 4.4: Foot trajectory in the sagittal plane during forward walking.

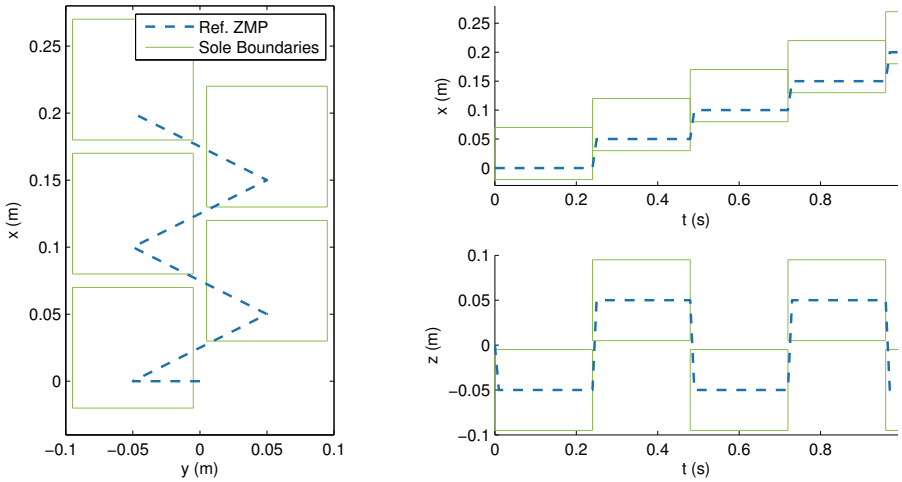
$$\frac{{P_3}^j - {P_2}^j}{T^j} = \frac{{P_1}^{j+1} - {P_3}^j}{T^{j+1}} \tag{4.3}$$

Since the initial and the final position of the trajectory are defined by the Footstep Planner, they will provide the points ${P_0}^0$ and ${P_3}^n$. In the same way, we know that the nonsupporting foot that performs the trajectory was the supporting foot in the previous and in the following step, so it must be still at the beginning and at the end positions. This forces the 0 value for the points ${P_1}^0$ and ${P_2}^n$. The rest of the points of the trajectory and the duration each one of the Bézier curves can be set freely, as long as constraints (4.2) and (4.3) are true.

4.5 CoM Trajectory Design

A paramount contribution to balance in walking robots arises from the movement of the CoM. Although the design of trajectories for the CoM remains an open issue, the most popular approaches for robust pattern generation are based on the ZMP stability criterion applied to an inverted pendulum model. The drawback of the use of such combination is the neglect of the rotational inertia of the robot. This assumption is acceptable in controlled environments, where external disturbances are limited, but general applications involve the presence of unexpected collisions. For this reason, we employ the ZMP criterion for the feedforward control of the CoM. It provides a base trajectory that will be modified by the feedback controllers to adapt to disturbances or unexpected states of the system.

The ZMP stability criterion states that in order to prevent the robot from tipping over, it is necessary to keep the ZMP within the support polygon or the convex hull of the robot. To simplify the calculation of the position of the ZMP, we will employ the Linear Inverted Pendulum Mode (LIPM). The LIPM involves two assumptions. First, the robot behaves like a single point mass concentrated



(a) Transversal plane. (b) Temporal evolution.

Figure 4.5: The ZMP reference trajectory jumps in every step to the ankle projection on the ground.

at the center of the mass distribution of the body. And second, the motion of the point mass is restricted to a horizontal plane.

Before generating the trajectory of the CoM, a set of future footsteps must be prepared. The number of footsteps to be calculated depends on the preview interval, that is, the period in the future considered to optimize the balance criteria. The sequence of the future footsteps defines the evolution that will follow the convex hull, hence it is possible to set a reference trajectory for the ZMP. A logical choice is to set the ZMP at the center of the convex hull at any instant. During the single support phases, the convex hull is the sole of the support foot and, in this work, we assume that there is no double support phase. Nonetheless, in humanoid robots the ankle is closer to the heel than to the toe. Setting the ZMP at the center of the sole would involve a constant extra load to the ankle joints. For the Nao robot, such a configuration would overheat the ankle motors in a few minutes. Therefore, a common choice is to sacrifice robustness in exchange of reasonable operational time. Hence we set the ZMP at the vertical projection of the ankle of the support foot on the ground, which provides the target trajectory for the ZMP during the next steps. Figure 4.5 shows the ZMP trajectory designed for a sequence of footsteps that corresponds to a forward motion.

At this point, the information required to generate the movement of the CoM is ready. This trajectory will perform an accurate tracking of the reference ZMP

trajectory. According to the LIPM model, the relation between the ZMP and the state of the CoM is simply given by:

$$\ddot{x} = \frac{g}{z}(x - p), \tag{4.4}$$

where x and z are the coordinates of the CoM in the sagittal plane, p is the position of the ZMP and g is the gravitational constant. The expression is also valid for the frontal plane and both dimensions are separable. In this way, given a reference trajectory for the ZMP, there is always an exact solution for the trajectory of the CoM, but it can easily diverge. In this work, we will use a Model Predictive Controller (MPC) to obtain a trajectory of the CoM that minimizes both the tracking error of the reference ZMP trajectory and the square of the control signal.

The version of the Model Predictive Controller implemented in this work was proposed by Wieber in [105]. It provides an efficient closed-form solution to the optimal control problem over a preview interval. The control signal is the jerk (first derivative) of the CoM, which keeps constant during the control cycle. The state of the system comprises the position, velocity and acceleration of the CoM. The state equation becomes:

$$\begin{bmatrix} x_{k+1} \\ \dot{x}_{k+1} \\ \ddot{x}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ \dot{x}_k \\ \ddot{x}_k \end{bmatrix} + \begin{bmatrix} T^3/6 \\ T^2/2 \\ T \end{bmatrix} \dddot{x}_k. \tag{4.5}$$

The cost function is a weighted combination of the sum of squared ZMP tracking errors and the sum of the squared jerks over the preview interval:

$$\min_{\dddot{x}_k, \dddot{x}_{k+1}, \ldots} = \sum_{i=k}^{\infty} \frac{1}{2} Q \left(p_{i+1} - p_{i+1}^{ref} \right) + \frac{1}{2} R \dddot{x}_i^2 \tag{4.6}$$

where R and Q are the weights for the reference tracking error and minimum jerk respectively.

The base idea of Model Predictive Control approaches is to calculate the optimal trajectory for a certain state and to repeat the process for the next cycle with an updated measure of the state. In this way, only the first sample of the CoM trajectory obtained need to be calculated, because the rest will be ignored. The influence of a sample of the reference trajectory of the ZMP on this sample of the CoM decreases with the time interval between them. Therefore, we can establish a limit for the reference trajectory of the ZMP, without affecting significantly to the following position of the CoM. Equation (4.7) applies a limit

of N samples to the trajectory of the ZMP:

$$\min_{\dddot{x}_k,\ldots,\dddot{x}_{k+N}} = \sum_{i=k}^{k+N-1} \frac{1}{2} Q \left(p_{i+1} - p_{i+1}^{ref} \right) + \frac{1}{2} R \dddot{x}_i^2. \tag{4.7}$$

Following the approach described in [105], we employ a compact notation and exploit recursively to transform the Quadratic Program and obtain a simpler formulation:

$$\min_{\dddot{\boldsymbol{X}}_k} = \frac{1}{2} Q \left(\boldsymbol{P}_{k+1} - \boldsymbol{P}_{k+1}^{ref} \right) + \frac{1}{2} R \dddot{\boldsymbol{X}}_k^2, \tag{4.8}$$

where $\boldsymbol{P}_k$ and $\boldsymbol{X}_k$ are the compact representation of the set of N samples of p and x respectively:

$$\boldsymbol{P}_k = \begin{bmatrix} p_k & \ldots & p_{k+N-1} \end{bmatrix}_{1\times N}. \tag{4.9}$$

$$\boldsymbol{X}_k = \begin{bmatrix} x_k & \ldots & p_{k+N-1} \end{bmatrix}_{1\times N}. \tag{4.10}$$

Equation (4.8) can be solved analytically, leading to:

$$\dddot{x}_k = -e \left(\left(\boldsymbol{M}_u^T \boldsymbol{M}_u + \frac{R}{Q} \boldsymbol{I} \right)^{-1} * \boldsymbol{M}_u^T \left(\boldsymbol{M}_x \hat{x} - \boldsymbol{P}_k^{ref} \right) \right). \tag{4.11}$$

where $\boldsymbol{I}$ is the identity matrix of order N and the matrices $\boldsymbol{M}_u$ and $\boldsymbol{M}_x$ are defined as:

$$\boldsymbol{M}_u = \begin{bmatrix} \frac{T^3}{6} & 0 & 0 \\ \vdots & \ddots & 0 \\ (1+3N+3N^2)\frac{T^3}{6} & \ldots & \frac{T^3}{6} - T\frac{z}{g} \end{bmatrix}_{N\times 3}, \tag{4.12}$$

$$\boldsymbol{M}_x = \begin{bmatrix} 1 & T & \frac{T^2}{2} - \frac{z}{g} \\ \vdots & \vdots & \vdots \\ 1 & NT & \frac{N^2T^2}{2} - \frac{z}{g} \end{bmatrix}_{N\times 3}, \tag{4.13}$$

Finally, $\boldsymbol{e}$ is a unit vector that selects the first element of $\boldsymbol{X}_k$:

$$\boldsymbol{e} = [1, 0 \ldots 0]_{1\times N}, \tag{4.14}$$

In every control cycle only the state of the CoM in the next cycle is calculated. The position of the CoM is delivered to the inverse kinematics module and the state of the CoM is used as input for the next time step.

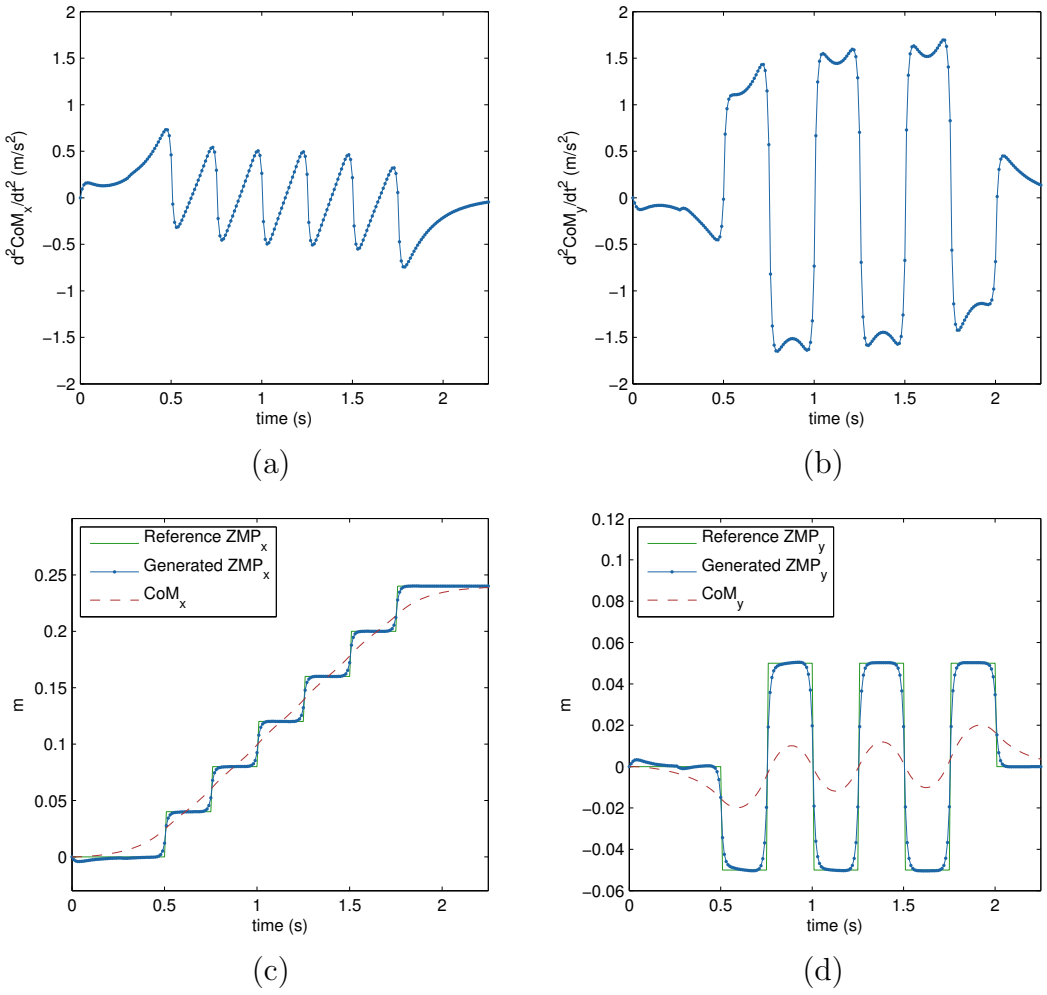


Figure 4.6: ZMP tracking performed by MPC in sagittal and frontal planes.

Figure 4.6 shows the performance of the MPC for a robot that is initially standing still and takes six steps before stopping again. As previously mentioned, the reference trajectory for the ZMP jumps abruptly from a fixed point on the sole of one of the feet to another fixed point o n the other. However, the proper configuration of the R/Q ratio softens this transition of the ZMP, reducing the sharp increments required in the acceleration of the CoM. In Figure 4.6(a) and Figure 4.6(b) the acceleration applied to the CoM is shown. Both in the sagittal and in the frontal plane, the acceleration of the CoM takes at least four cycles to perform the transition. On the other hand, Figure 4.6(c) and Figure 4.6(d) display the trajectory of the reference ZMP, the trajectory obtained by the MPC and the trajectory of the CoM. One can observe that the trajectory of the ZMP

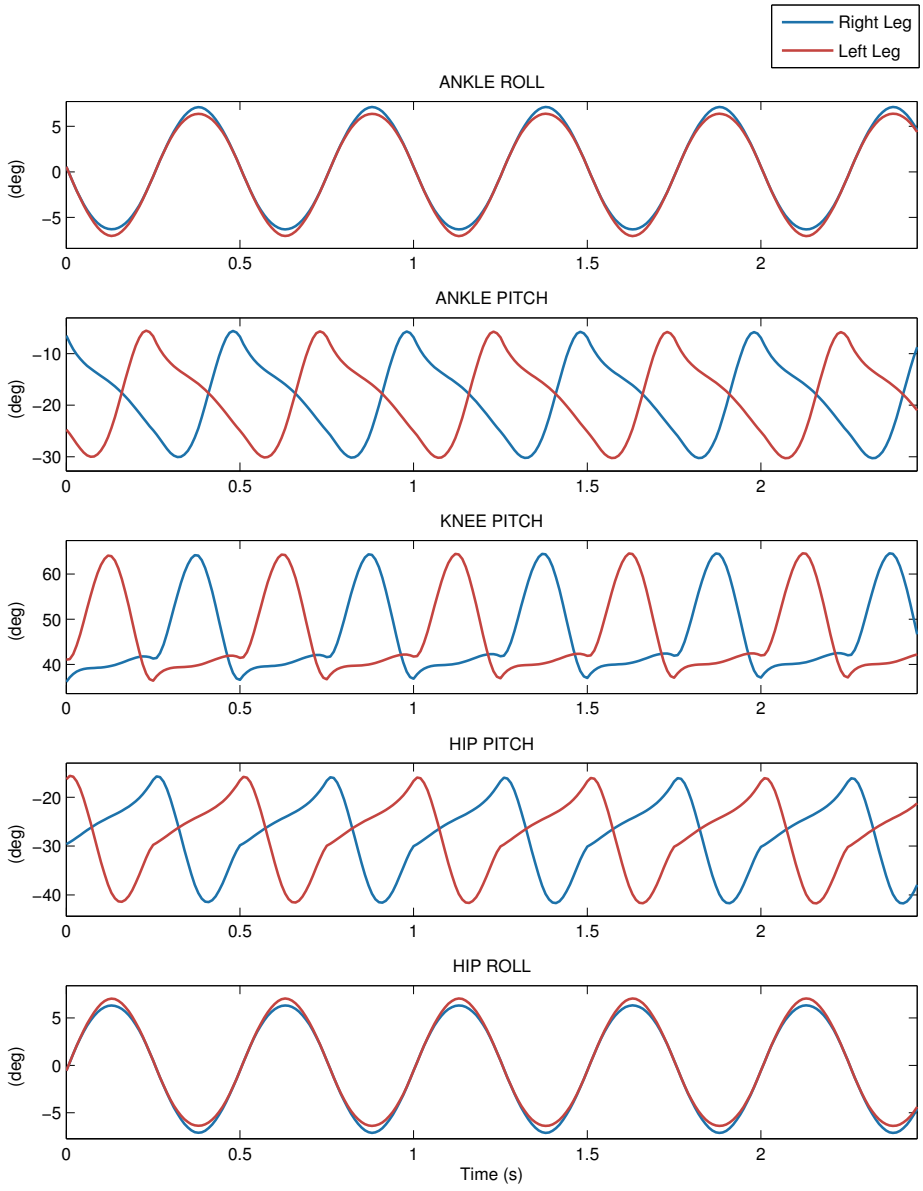


Figure 4.7: Walking pattern in joint space.

is accurately tracked both in sagittal and in frontal planes.

Figure 4.7 shows, on the other hand, the evolution of the joint positions resulting from a prolonged forward walking motion. That is, the output of the *Inverse Kinematics* Module. It should be noticed that the minimization of the

jerk in the trajectory of the CoM directly translates to a smooth evolution of the positions of the joints. This feature is a requirement for balance and involves additional advantages such as longer battery life and reduced hardware wear. Finally, Figure 4.8 illustrates the omnidirectional capabilities of the approach by showing the CoM and ZMP trajectories generated for a circular motion.

To provide the walking pattern generator with the information required to generate the reference trajectory of the ZMP during the preview interval, several footsteps must be scheduled in advance. Nevertheless, if an elevated number of footsteps is scheduled, the reactivity of the walking pattern generator is penalized. For that reason, we only queue new footsteps 0.5 seconds in advance. To fulfill the rest of the preview interval, we employ a set of artificial footsteps that replicates the same velocity than the last calculated footstep.

Table 4.1 shows the typical values for the parameters of the MPC, which can be finely tuned according to specific needs. In this work, these adjustments are made empirically, and the same values are employed for sagittal and lateral components.

Param.	Value
T	10 ms
N	100
CoM_z	260 mm.
R/Q	10^{-7}

Table 4.1: Parameters of MPC.

The value of T is set to the duration of the control cycle of the middleware provided by the manufacturer. On the other hand, the product NT defines the duration of the preview interval. In [105] is demonstrated that the minimum value of N to ensure the stability of the control scheme depends on the R/Q ratio. Additionally, the maximum value of N is limited by the computing time available for walking pattern generation. The height of the CoM is constrained by the kinematics of the robot. Lower values of the CoM involve a high load to the pitch joints of the legs and require elevated accelerations of the CoM to manipulate the ZMP. The value of the R/Q ratio sets the trade-off between accurate tracking of the ZMP and reduced jerk of the CoM. Since the use of 3D-LIPM involves strong simplifications and the walking pattern generator is complemented by a control system, accuracy of ZMP tracking can be slightly relaxed.

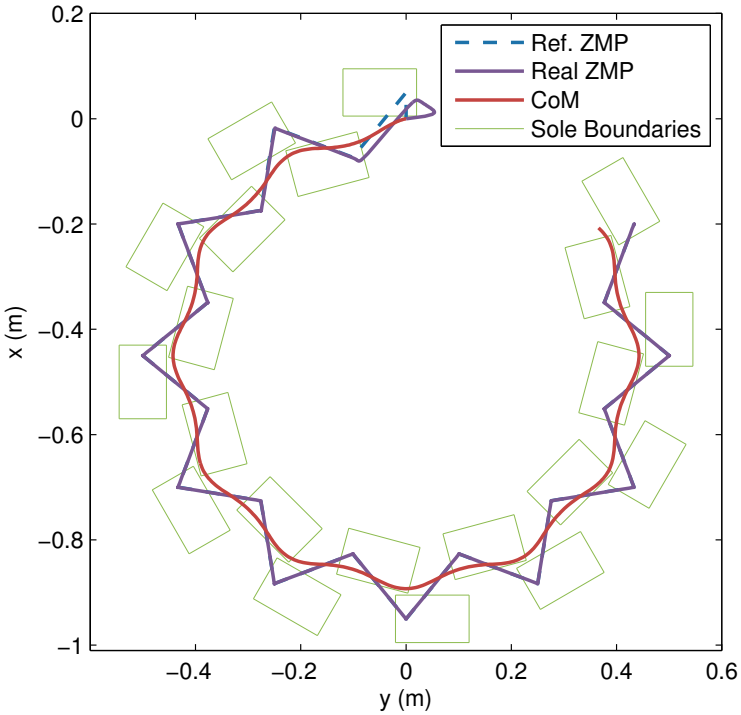


Figure 4.8: CoM and ZMP trajectories of a circular motion.

4.6 Discussion

In this chapter we have described a set of approaches to generate the walking pattern for the robot. The goal of this pattern is to provide a base for the locomotion of the robot that is stable under ideal conditions (accurate representation of the model and no external disturbances). Although the output of the walking pattern algorithms is the pose of the robot in the com-torso-foot space, we focused on the calculation of the position of the CoM and the trajectory of the swing foot. On the other hand, the orientation of the torso is kept vertical and its impact on the balance is not studied in this work.

The walking pattern generator approach that we have selected for this work is based on the preview control of the ZMP and the 3D-LIPM model. The reason to use this approach instead of a central pattern generator is that it uses a kinematic model for the robot. The logical structure of model-based approaches allow faster identification of problems or potential enhancements compared to black-box approaches like CPGs.

Nevertheless, the range of walking pattern generation approaches that use a kinematic model of the robot offers several options. The Resolved Momentum Control approach provides the most complete solution to the problem of generating trajectories for the limbs that direct the robot with a certain criterion while maintaining balance. This approach takes into consideration both the linear and the angular momentum of the robot and also the contacts with the environment. However, the prominent computational requirements and the impossibility to specify the trajectories of the foot took us to discard this approach.

The use of analytic trajectories based on the 3D-LIPM equations allows model-based low computing requirements design of trajectories but also involves some difficulties. The trajectories generated by the 3D-LIPM can be linked together in two ways: either allowing a transition stage during the double support phase where the balance is not ensured or renouncing to the control of location and duration of the footsteps.

Since it is our interest to maintain the control of the location and duration of the footsteps, the model predictive control of the ZMP is the approach that suits better to our needs. The computational requirements of this approach are halfway between those of resolved momentum control and the analytic trajectories based on the 3D-LIPM.

To generate the walking pattern is therefore necessary to plan the future footsteps. The target location for the footsteps depends on the type of motion required for the robot. However, their feasibility is conditioned by the kinematic configuration of the legs and limited by the collision between both feet. We have presented in this chapter an empirical solution to obtain an area of reachable

locations for the leg and an analytic approach to prevent collisions between the feet.

Given the locations and duration of a sequence of steps, we opted for a simple solution to generate the trajectory for the ZMP, placing its reference location at the center of the dynamic limits of the support polygon. This trajectory is processed by the model predictive controller to generate the trajectory for the CoM that tracks its evolution. In this way, we obtain the location of the CoM that will be delivered to the inverse kinematics module. On the other hand, to calculate the trajectory of the swing foot that takes it from one footstep location to the next, we employ a cubic Bézier spline. This curves provide smooth trajectories and ample flexibility in the design of the trajectories.

Finally, we have shown the walking patterns generated for straight and curved motions. The resulting trajectories allow the robot to move omnidirectionally while providing the base for a well-founded balanced gait.

Chapter 5

Balance Control

5.1 Introduction

The approach described in the previous chapter to generate the walking pattern for the robot provides a good base to achieve a balanced gait. It generates smooth trajectories for the CoM that project gravity and inertial forces over the support soles. However, this is not sufficient to obtain a robust gait. The model employed assumes that the soles are flat on the ground and the actuators perfectly track the reference positions. Moreover, it does not account for disturbances and ignores the rotational inertia of the parts of the robot. In this section we analyze the problem caused by the rotational dynamics and we describe two feedback controllers that can counteract the simplifications of the model to generate a robust gait on a physical robot. Additionally we provide details on the sensor filter implemented to prepare the information required by the controllers. The resulting architecture of the system once the sensor feedback is included is depicted in Figure 5.1.

The first controller that we present deals with the rotational dynamics of the robot, which is important both in the sagittal and in the frontal plane. The second controller is specific for the frontal plane. In the frontal plane the position of the CoM oscillates between both feet, inverting the sign of the speed of the CoM in every step. The sign in the lateral acceleration of the CoM depends on the foot that is used to impulse the CoM, which is the support foot. Therefore, an appropriate strategy to improve the balance is to adapt the instant when the role of the support foot alternates from one foot to the other.

5.2 Sensors Processing

The feedback controllers that will be proposed in this chapter require the use of the sensors of the robot to estimate the pitch and roll angles between the sole

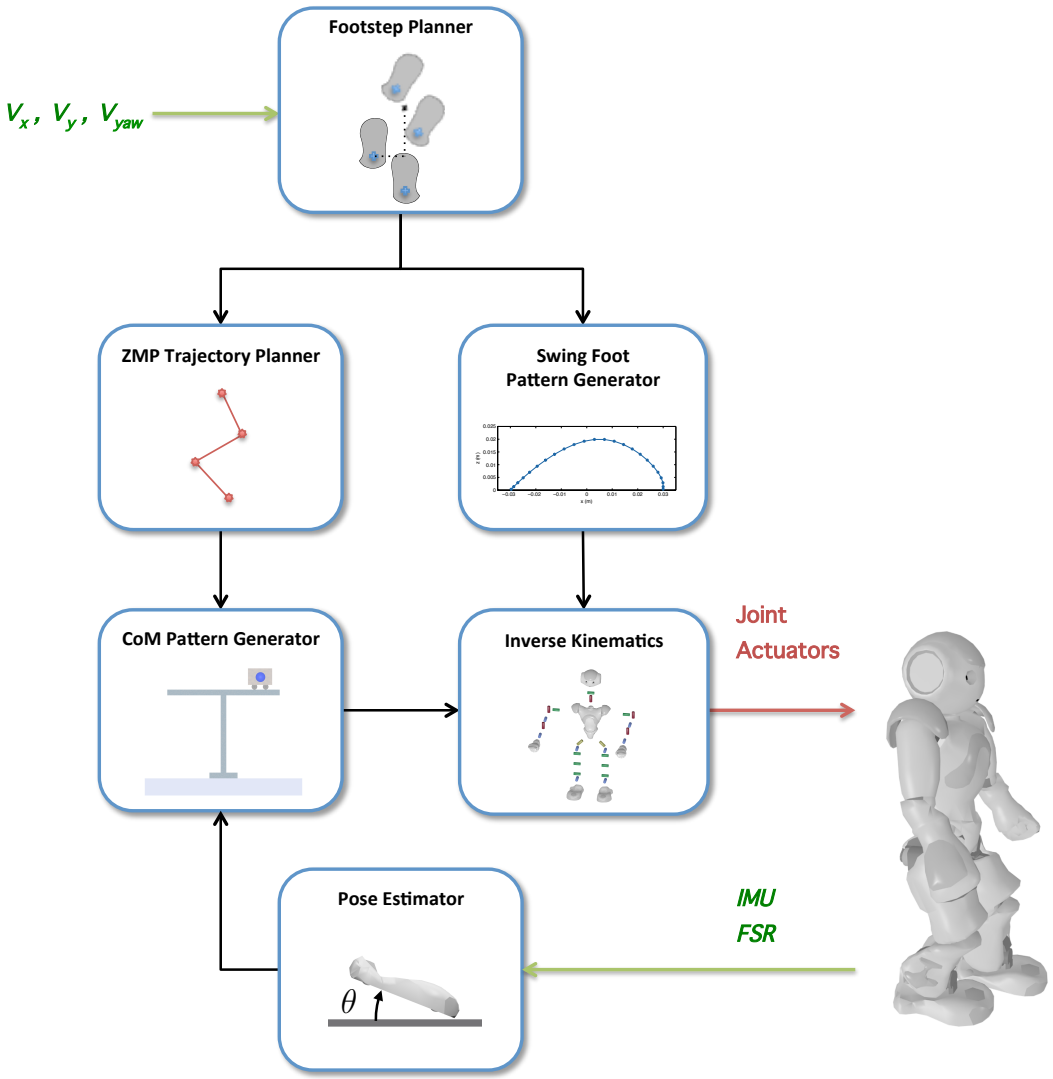


Figure 5.1: Closed-loop software architecture for locomotion system.

and the ground. The Nao robot has several types of sensors that provide useful information for balance. The accelerometers and the gyrometers of the inertial unit can be employed to estimate the attitude of the torso, the joint sensors inform about the real position of the CoM in the sole frame, and the Force Sensitive Resistors (FSR) of the soles could theoretically indicate the position of the Center of Pressure (CoP) and the magnitude of the ground reaction force.

Nevertheless, the noise of the FSRs measurements makes the estimation of the CoP very difficult. Hence we will use them only as contact detectors to determine if the soles are in contact with the ground. The joint sensors, on the other hand, provide quite accurate estimations of the backlash of the robot. However, we are not dealing with the backlash effect explicitly in this work. We will focus on the stability against external disturbances, whose effect is better measured by the sensors of the inertial unit. Therefore, we have developed a Kalman filter that fuses measurements provided by accelerometers, gyrometers and FSRs to estimate the orientation of the torso. Since we set the pose of the torso perpendicular to the sole of the support foot, we assume that the angles between the torso and the vertical axis are equal to the angles between the sole and the ground.

The state of the Kalman filter consists of the angle between the ground and the sole and its angular velocity. Once again, the sagittal and lateral components are calculated separately. Since the procedure is similar in both planes, we will only detail the equations for the frontal plane, using ρ for the roll angle between the sole and the ground and $\dot{\rho}$ for the roll velocity. For the definition of the state transition model, we assume that the rotational velocity of the sole angle is constant, and the rotational acceleration is included in the process noise ($\boldsymbol{w}$):

$$\begin{bmatrix} \rho_{k+1} \\ \dot{\rho}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \rho_k \\ \dot{\rho}_k \end{bmatrix} + \boldsymbol{w}_k. \tag{5.1}$$

The roll and pitch velocity provided by the sensors are then employed to update the estimated state. The update is performed as stated in equation (5.2), where $\boldsymbol{v}_k$ is the measurement noise.

$$\begin{bmatrix} \hat{\rho}_k \\ \hat{\dot{\rho}}_k \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \rho_k \\ \dot{\rho}_k \end{bmatrix} + \boldsymbol{v}_k \tag{5.2}$$

For the estimation of the roll and pitch velocities, the only requirement is to remove from the raw gyrometer measurements the zero offset. For the estimation of the attitude of the torso, on the other hand, a different procedure is followed in the sagittal and frontal planes. While in the sagittal plane only the accelerometers are used, in the frontal plane both the accelerometers and the FSR measurements are employed.

To estimate the attitude of the torso with the accelerometers, we calculate the direction of the total force measured and assume that this force is the gravity.

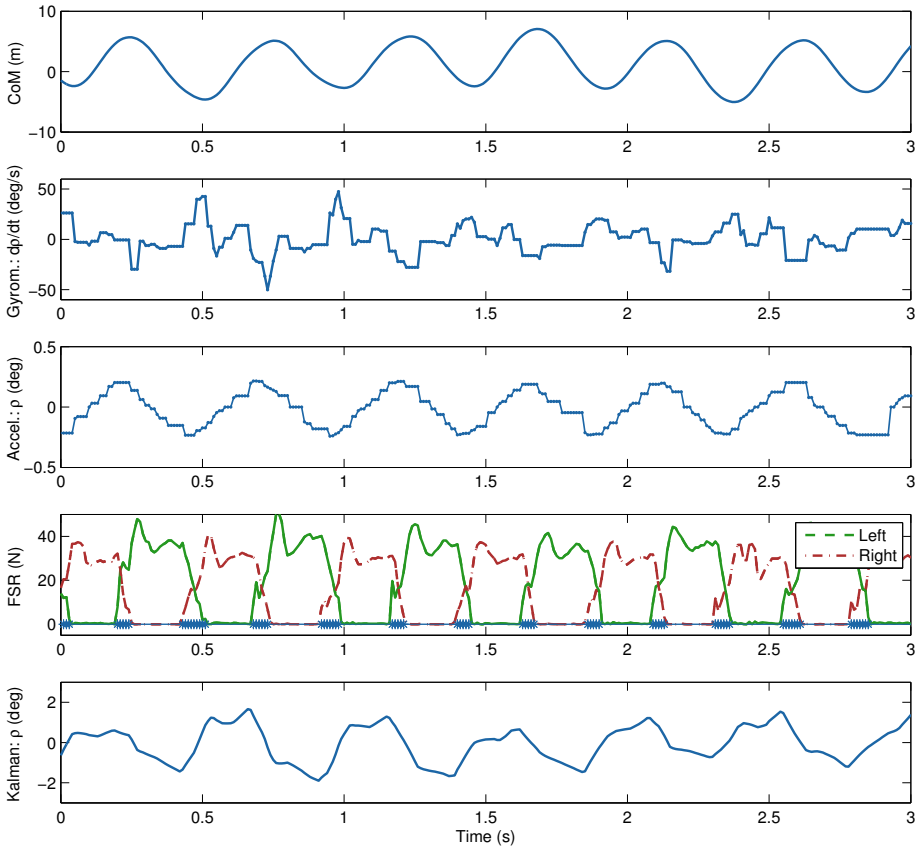


Figure 5.2: Sensor fusion performed by the Kalman filter to estimate the sole roll angle.

Since the direction of the gravity force is known, it is possible to obtain pitch and roll angles of the torso. Nevertheless, in this way we are neglecting the accelerations applied to the torso by the walking pattern generator. Since the duration of the steps is reduced, the error in the attitude estimation caused by the walking pattern generator has higher frequency than the real torso oscillations. Therefore, by setting a sufficiently high value for the noise level of the attitude estimation, it is possible to filter out the error in the attitude estimation generated by the walking pattern generator.

Although this configuration works adequately in the sagittal plane, in the frontal plane the oscillatory nature of the CoM trajectory involves high accelerations and the attitude error accumulated by the integration of the gyrometer measurements cannot be stabilized by the accelerometer attitude estimations. To solve this problem, we make the assumption that if a contact is detected in both feet simultaneously, the roll angle of the sole is close to zero. In this way, when a simultaneous contact is detected, the roll estimation, $\hat{\rho}$, provided by the accelerometers is replaced by zero and the attitude sensor noise is reduced.

Figure 5.2 illustrates the performance of the filter in the frontal plane during regular forward walking. The first row simply displays the oscillatory trajectory of the CoM to allow relating events in the sensor signals with the corresponding walking phase. The second plot shows the roll velocity estimated by the gyrometer. The attitude estimated from the accelerometers is displayed in the third row. Next, the forth row shows the total force measured by the FSRs of each foot. In the same row, the blue starts indicate that simultaneous contact is detected. Finally, the bottom row depicts the roll angle estimated by the Kalman filter.

5.3 Angular Momentum Control

In this section we introduce the controller that permits to improve the rotational stability of the robot. In the previous chapter we described a walking pattern generator that is based on an open-loop 3D-LIPM model. Below we will analyze the problems derived from the use of this model. Since the planar motion restriction of this model is frequently ignored, we propose a new model that explicitly considers this fact: the Tilting Base Inverted Pendulum (TBIP). With the help of the new model we will analyze the rotational dynamics of a simple case: a disturbed steady robot. Finally, we will make use of the TBIP to design a simple controller and analyze its effect on the rotational stability of the robot through simulations.

5.3.1 Walking Pattern Generator Analysis

The walking pattern generator described in the previous chapter bases the tracking of the reference trajectory of the ZMP on the 3D-LIPM model. This approach involves severe simplifications that have an impact in the performance of the walking pattern generator. Let us analyze them:

- **Open-Loop**. The most obvious problem that the aforementioned walking pattern generator exhibits is the absence of a control feedback. As soon as the system deviates from its planned state because of error accumulation or the action of external forces, the evolution predicted by the model becomes no longer valid.

- **Inverted Pendulum**. As an inverted pendulum based model, the mass of the whole robot is assumed to be concentrated at a single point. This point is connected to the ground by a telescopic massless leg. Since the mass of the whole robot is concentrated at a point, the moment of inertia around that point becomes zero and the centroidal angular momentum is neglected. However, in real robots the masses of the legs can attain a significant proportion of the total mass of the robot. For instance, in the robot Nao, the mass of each leg accounts for 20 % of the total mass. When the legs are moved at a high velocity to perform quick steps, the magnitude of the error made is considerable.

- **ZMP**. While the ZMP is an excellent tool to plan the motion of the robot and generate balanced walks, its efficacy to counteract external disturbances is more controversial. First, the area of the support polygon dramatically reduces when the soles are tilted. In this way, right after the perturbation the location of the ZMP would have to jump to an accurate location. This involves highly performing sensors and actuators that are not always present in robotics platforms. Moreover, it should not be forgotten that the existence of a ZMP ensures that there is no moment affecting the movement of the system. In order words, it ensures that no rotational acceleration will be applied to the body of the robot. In this way, in case the robot is already rotating because of the action of an external disturbance, the existence of a ZMP will ensure that the body continues its rotation at a constant rotational velocity, which is not a reasonable behavior to avoid the fall.

- **Planar movement** In order to linearize the motion equations, the 3D-LIPM imposes the planar movement restriction for the CoM. Specifically, we used the horizontal plane for our walking pattern generator.

Since the soles were assumed to lay perfectly flat on the ground, it was sufficient to locate the CoM in the plane parallel to the reference sole. However, when the robot receives an external disturbance the angle between the sole and the ground varies, and the movement of the CoM is no longer horizontal or even planar.

5.3.2 Tilting Base Inverted Pendulum

Let us now analyze the effect of the dynamics of the CoM on the pose of the sole of the robot. For that, we will assume that the friction of the ground is infinite and any of the extremes of the sole can act as a passive joint when the foot is tilted. Additionally, we will maintain the restriction that limits the movements of the CoM to a plane which is parallel to the sole of the support foot.

If we assume a point mass robot placed at $\boldsymbol{C}(x, y, z)$ and a reference frame located at the projection of the ankle on the sole of the support foot, the torque is given by:

$$\boldsymbol{\tau} = \boldsymbol{r} \times \boldsymbol{F}, \tag{5.3}$$

being $\boldsymbol{r}$ the distance from the rotation axis to the CoM and $\boldsymbol{F}$ the force applied on it. The rotation axis is located at one of the extremes of the sole. For simplicity, we will study the dynamics of the robot in the sagittal plane, but results are similar in the frontal plane. In this way, the inertial and gravitational forces acting on the sagittal plane of the robot are:

$$F_x = -M\ddot{x} + MG\sin\theta \tag{5.4}$$

$$F_z = -MG\cos\theta, \tag{5.5}$$

where θ is the angle between the sole and the ground around the y axis located at the limit of the sole x_b, M is the mass of the robot and G is the gravitational constant. The torque around x_b resulting from these forces is

$$\tau_\theta = M\left(G(z\sin\theta - x_b\cos\theta + x\cos\theta) - \ddot{x}z\right). \tag{5.6}$$

It should be noticed that the acceleration of the CoM , $\ddot{x}$, has a significant impact in the amount of angular momentum of the robot, which is:

$$L = I\dot{\theta}. \tag{5.7}$$

At this point another simplification is made to the model. In regular configurations for the Nao robot, the range of x is limited to $[-30, 40]$ mm and the height of the CoM is fixed around 260 mm. Therefore, the moment of inertia,

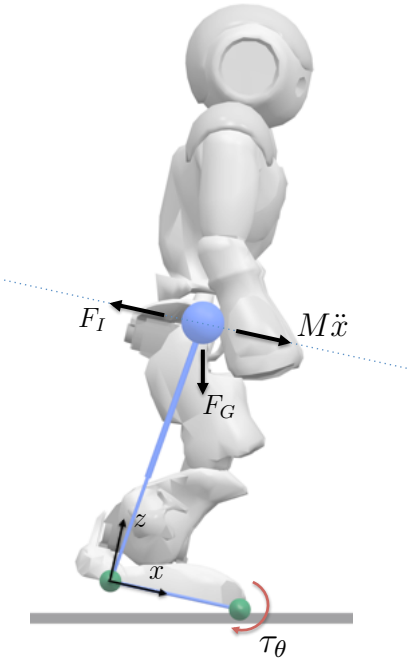


Figure 5.3: Tilting Base Inverted Pendulum.

I, of the robot around the border of the sole does not change significantly during operation and we will consider it constant. In this way, we can state that the angular acceleration $\ddot{\theta}$ is approximately proportional to the torque around the border of the sole x_b.

5.3.3 Case Study: Disturbed Steady Robot

According to the walking pattern generation approach described in the previous chapter, the movement of the CoM is restricted to a horizontal plane of constant height. Since we assumed that the soles of the robot were firmly flat on the ground, the trajectories of the CoM delivered to the inverse kinematics module were expressed in a reference frame located on the sole plane.

However, the oversimplified dynamical model used by the Model Predictive Controller (MPC) and the unexpected collisions with elements of the environment cause the inclination of the sole plane. Under these circumstances, the convex hull of the robot reduces to the thin line or to the point were the boundaries of the sole contact the ground. In this way, the point where the projection of the sum of inertial and gravity forces acting on the CoM intersects the ground will be out of the limits of the support polygon, and will generate a tipping moment on the robot. The point that we called ZMP becomes therefore a FZMP (Fictitious

Zero Moment Point). Moreover, since the plane of the sole is tilted, the real state of the CoM will not correspond to the internal state of the MPC, which is referenced to the sole frame. Because the CoM position used by the inverse kinematics module is obtained from this state, the location of the FZMP will not match the ZMP position calculated by the MPC.

For instance, if the target position for the ZMP is set to a fix point located between the ankles of the robot, the trajectory of the CoM obtained by the MPC will converge to the vertical projection of the ZMP over the constant height plane. In the absence of feedback control, this position will be maintained regardless the external disturbances. In this situation, if the robot suffers a disturbance, the sole plane will tilt and the FZMP will move away from the location of the ZMP. If the magnitude of the disturbance is reduced, the torque generated by the FZMP on the limit of the convex hull will reduce the rotational velocity of the robot until, eventually, the rotational velocity of the sole plane will invert its sign and the robot will start moving to recover the horizontal position. Once the sole recovers the horizontal position, however, the magnitude of the rotational velocity has grown and the robot has accumulated a certain angular momentum that will make it rotate in the opposite sense, and the whole process is repeated.

For a perfectly rigid robot, these oscillations will repeat perpetually, alternating the rotation axis from one extreme to the other of the convex hull. In real robots, however, the limbs are flexible and the motors have finite torque, which dampens the oscillations. If a still standing Nao robot is tilted laterally and released free, the oscillations are visible with the naked eye for a few seconds. In this way, although the generation of trajectories for the CoM using the walking pattern generator of the previous section is a good mechanism to prevent the rotations of the robot body, it does not favor the reduction of the rotational velocity in the system when it appears.

The goal of the angular momentum controller that we propose is to modify the trajectory of the CoM obtained by the MPC to adapt the torque $\boldsymbol{\tau}$ generated at the limits of the convex hull. The torque is modified to favor the reduction of the angular velocity. This idea is illustrated in Figure 5.4, where the extra acceleration induced by the angular momentum controller generates an Inertial Force, F_I, that shifts the position of the FZMP according to the rotational speed.

5.3.4 Controller Design

In (5.6) we appreciated a direct contribution of the CoM acceleration $\ddot{x}$ to the torque around the limit of the sole. In order to design a controller that can reduce the angular momentum of the system, it seems logical to use $\ddot{x}$ for the control action of the controller. However, the acceleration of the CoM is already managed by the model predictive controller that we introduced in the previous chapter.

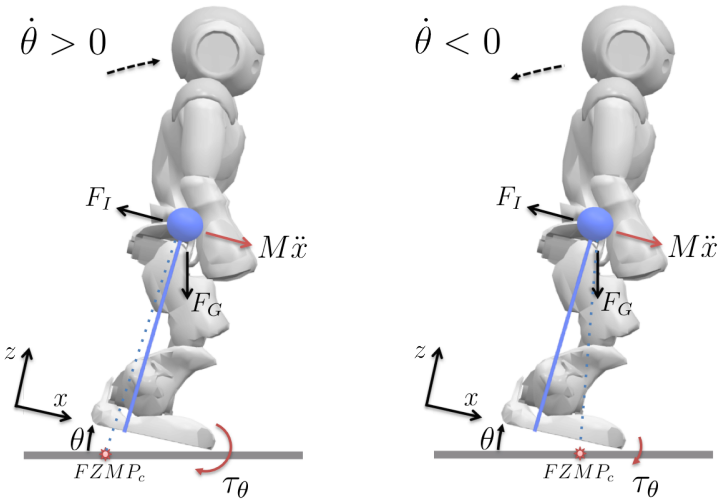


Figure 5.4: The acceleration of the CoM generated by the angular momentum controller affects to the restoring moment of the sole of the robot.

Since we want to keep a reasonable trajectory for the CoM when the sole lays flat on the ground, the angular momentum controller will simply add an acceleration term ($\ddot{x}_c$) to the acceleration calculated by the angular momentum controller. In order to counteract the rotational velocity of the robot around the border of the sole with the additional acceleration of the controller, we will simply implement a proportional controller:

$$\ddot{x}_c = P\dot{\theta}, \tag{5.8}$$

where P is a positive definite constant. After integrating both members of (5.8) and discretizing with control period T:

$$x_{c_{k+1}} = x_{c_k} + P\theta_k T. \tag{5.9}$$

The increment in the position of the CoM due to the angular momentum control is therefore:

$$x_{AM_k} = P\theta_k T \tag{5.10}$$

By relating the torque around the sole to the angular velocity of the robot we obtain a controller that effectively reduces the rotational speed of the robot and that has no effect if the rotational dynamics of the robot are stable. Figure 5.5 illustrates the performance of the controller in a simulated environment that implements the model described above. The simulation starts with a steady

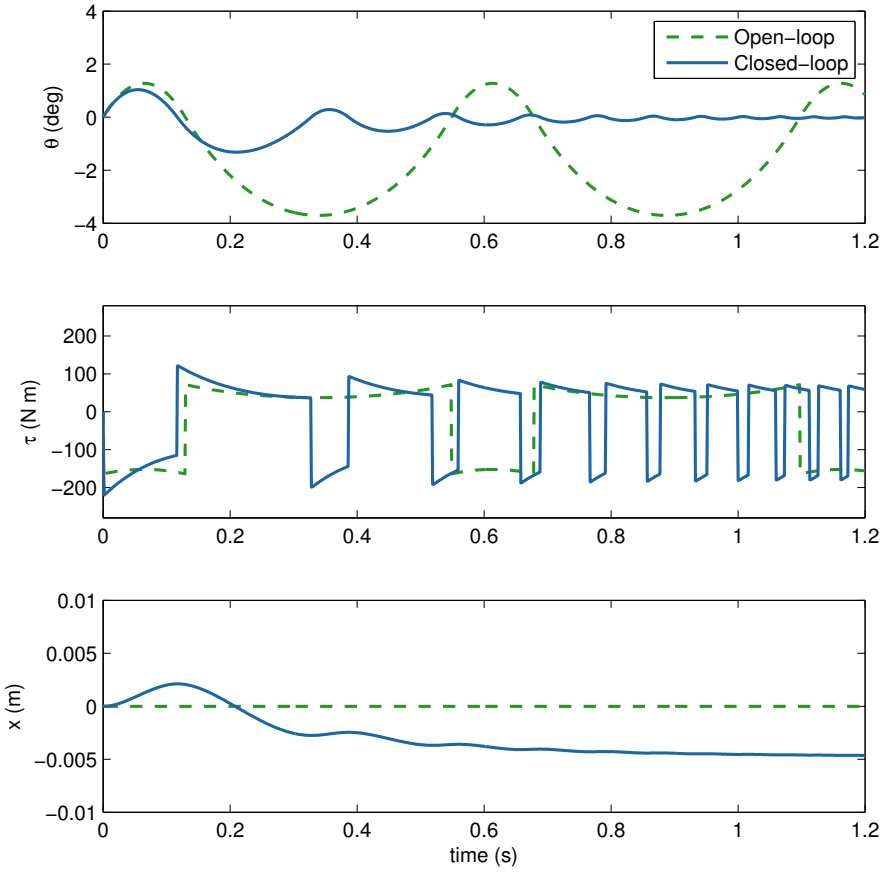


Figure 5.5: Simulated response of the angular momentum controller. At t=0 s a disturbance is simulated as a step increase in the angular speed of a still robot.

standing robot whose CoM position in the sagittal plane projected on the ground matches that of the ankles of the robot. At t=0s a disturbance is simulated as a step increase of the pitch velocity of the robot. Then, the evolution of an open loop system and a system with angular momentum control is compared.

In the first row, the oscillation of the pitch angle is perpetual in open-loop mode while it is considerably reduced under the action of the controller. The asymmetry in the oscillation is due to the different distance from the limits of the sole to the ankle. The second row shows the torque around the limit of the sole that is acting as rotating axis. It is interesting to notice that the magnitude of the restoring moment in the controlled system is higher than in the open-loop system at the beginning of every oscillation, and that it becomes fainter as the

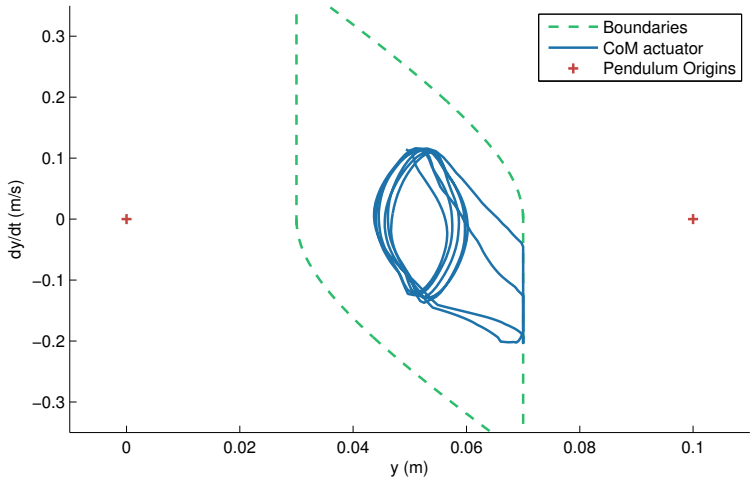


Figure 5.6: The states of the CoM delivered by the preview controller are bounded in the phase space. The plot shows a sample case when a robot walking on the spot is pushed twice and the trajectory of the CoM reaches the limits.

angle of the sole approaches the horizontal position again. The third row displays the sagittal position of the CoM in the sole frame.

When the robot is pushed from a side, the angular velocity controller will generate a yielding motion of the torso away from the pushing force to avoid the inclination of the sole. This absorption effect must be limited, however, because it can take the CoM to a position beyond the support foot, from where it is not possible to recover. To avoid this situation, the permitted CoM states are bounded in the phase space, as depicted in Figure 5.6.

As a side effect of the action of the controller there is a permanent offset in the position of the CoM at the end of the process, as can be appreciated in the third row of Figure 5.5. However, since the angular momentum controller is used in combination with the model predictive controller, as described before, the offset produced by the angular momentum controller is rapidly reduced. Figure 5.7 depicts the combination the angular momentum controller and the preview controller.

5.4 Timing Control

In Chapter 4 we described an approach to generate trajectories for the CoM that allows to keep the robot balanced while the feet execute a sequence of steps of a certain duration. The angular momentum controller modifies these trajectories

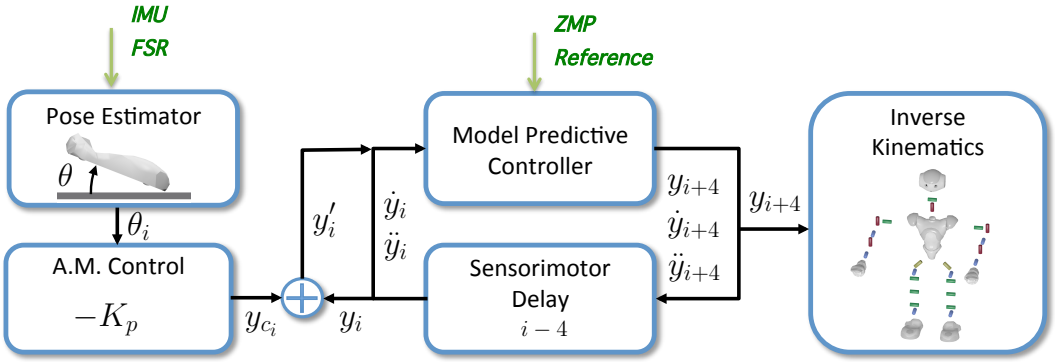


Figure 5.7: The angular momentum controller modifies the internal state of the Model Predictive Controller.

of the CoM to improve the balance when the robot is disturbed. However, when the action of the angular momentum controller is relevant, the state of the CoM will differ from the original trajectory planned by the MPC. Although the MPC updates the trajectory online to minimize the tracking error of the reference ZMP trajectory, for certain states the minimum error achievable is too high and the appearance of a tipping moment cannot be avoided.

Consequently, it is beneficial for balance to supervise the state of the CoM and to ensure that the footsteps pattern will not lead to an excessive tracking error of the ZMP reference trajectory. The footsteps pattern comprises both the location of the footsteps and their duration. As we said before, the adaptation of the instant when the role of the support foot is exchanged is a useful tool to control the trajectory of the CoM in the frontal plane.

Hence we propose here a timing controller that monitors the state of the CoM and modifies the duration of the steps if necessary. The goal of this controller is to ensure that the CoM trajectories generated by the MPC will not make an excessive ZMP tracking error.

To ensure a reduced ZMP tracking error we define an optimal CoM trajectory and aim at intercepting it at the exchange of support instant. The orbital energy of a trajectory is a concept borrowed from planetary mechanics that permits to define the trajectory of a CoM that follows the rules of the 3D-LIPM. If the base of the inverted pendulum is static, the orbital energy of the CoM is constant along its trajectory. The orbital energy is given by (5.11), where g is the gravitational constant.

$$E = \frac{1}{2}\left(\dot{y}^2 - \frac{g}{z}y^2\right) \tag{5.11}$$

Figures 5.8 and 5.9 show several constant energy orbits for adjacent support

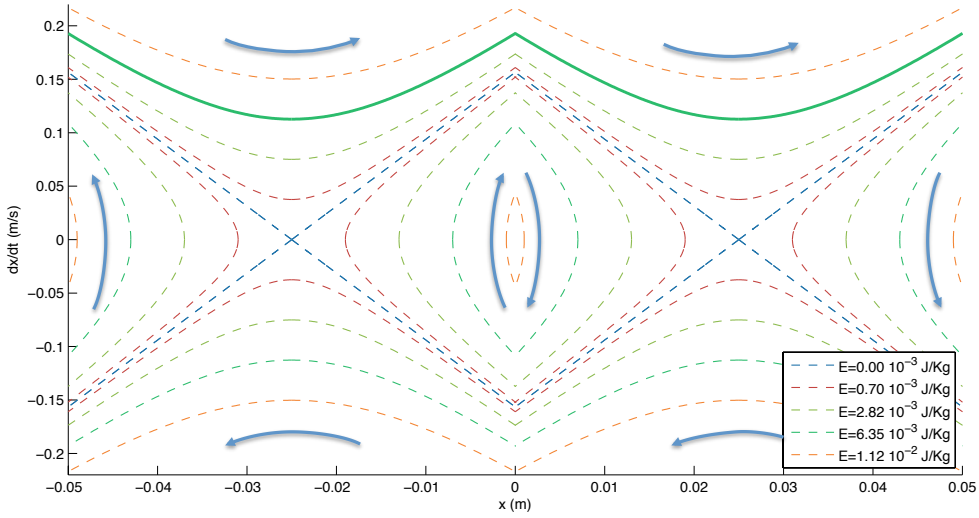


Figure 5.8: Energy orbits for sagittal component of movement during forward walking.

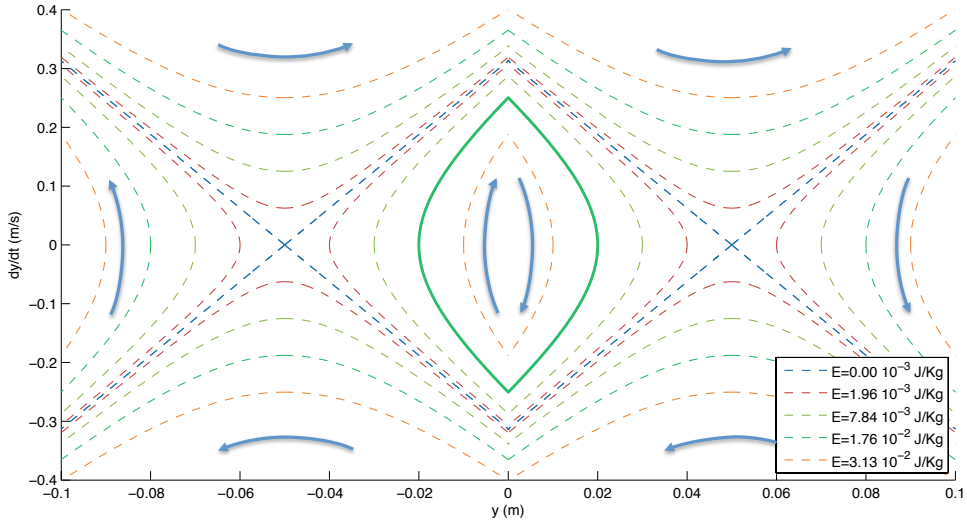


Figure 5.9: Energy orbits for lateral component of movement during forward walking.

points. Figure 5.8 represents a forward step size of 5 cm. The energy orbit followed by the CoM of a robot walking at a typical velocity of 15 cm/s is marked with continuous trace. In Figure 5.9 the feet are separated by 10 cm, that is, the distance that separates the hip extremes of both legs. In this case, the orbits do not escape the limits of the plot but are connected together conforming a cycle. The amplitude of the oscillations is determined by the duration of the step. The continuous trace, in this case, illustrates a step duration of 250 ms. It is important to notice that a sufficiently large deviation in the orbit can provoke the trajectory of the orbit to cross the region limits and cause the divergence of the state of the CoM.

To calculate the optimum step duration we will define a target orbital energy and calculate the time required by a CoM that is in a certain state to intercept that orbit from the point of view of the next support foot. For that, we need the equations of the 3D-LIPM:

$$y(t) = y_0 \cosh(kt) + \frac{\dot{y}_0}{k} \sinh(kt), \tag{5.12}$$

$$\dot{y}(t) = y_0 k \sinh(kt) + \dot{y}_0 \cosh(kt). \tag{5.13}$$

where y_0 and $\dot{y}_0$ are the position and velocity of the CoM at $t = 0$ and $k = \sqrt{g/z}$.

Since the robot cannot cross its feet in the frontal plane, the trajectory of the CoM must be oscillatory, that is, there will be an apex in the trajectory of the CoM during each step. If we set y_0 in that apex and t_0 at the instant where the apex is reached, equations (5.12) and (5.13) simplify as:

$$y(t) = y_0 \cosh(kt), \tag{5.14}$$

$$\dot{y}(t) = y_0 k \sinh(kt). \tag{5.15}$$

The current time and the apex position are calculated as

$$t_n = \frac{\operatorname{atanh}\left(\frac{\dot{y}_n}{y_n k}\right)}{k}, \tag{5.16}$$

$$y_0 = \frac{y_n}{\cosh(kt_n)}, \tag{5.17}$$

where y_n and $\dot{y}_n$ are the current estimated position and velocity of the CoM with respect to the center of the current support foot.

Provided the length of the step in the frontal plane S_y, we can calculate the orbital energy relative to the frame of the next support foot:

$$E_{sw}(t) = \frac{1}{2}\left(\dot{y}^2(t) - \frac{g}{z_c}(y(t) - S_y)^2\right). \tag{5.18}$$

Our goal is to change the support foot when E_{sw} has the value of the target orbital energy E_t. Substituting (5.14), (5.15), and (5.17) in (5.18), we can calculate the optimal instant t_s for the support exchange

$$t_s = \frac{\operatorname{acosh}\left(\frac{2E_t + k^2(S_y^2 + y_0^2)}{2y_0 k^2 S_y}\right)}{k}. \tag{5.19}$$

The remaining time to the optimal exchange instant will be $\Delta t_s = t_s - t_n$. Theoretically, one could use Δt_s to update the duration of the step. Nevertheless, the estimation of Δt_s is not accurate because the real trajectory that the CoM follows is not governed by the exact laws of the 3D-LIPM. In addition, reducing the duration of a step involves accelerating the trajectory of the stride, which is not convenient as we discussed in Chapter 4.

We have thus chosen to add only positive increments for a fixed duration of 10 ms (one control cycle) to the scheduled support exchange instant $\Delta t_{s_{sch}}$ when the difference with Δt_s exceeds the sensory motor lag. Furthermore, even though the MPC moves the ZMP along the sole of the support foot, we place the base of the inverted pendulum at the external limit of the sole to avoid excessive corrections.

Finally, we also implemented a measure to enforce regularity in the oscillations of the CoM. If the scheduled time for support exchange arrives before the CoM has covered a fraction (45%) of the distance between the current and the next pendulum origin, the scheduled time for support exchange is also delayed for one control cycle (10 ms).

5.5 Discussion

In this chapter we have presented a set of controllers to improve the balance of a biped. The first controller aims at reducing the rotational dynamics of the robot, and the second one supervises the feasibility of the commands that the locomotion system receives. The controllers are designed to be coupled to a ZMP-based walking pattern generator.

The combination of ZMP-based walking pattern generators and rotational dynamics suppression is a powerful approach to deal with the balance of the platform. ZMP-based balance approaches are well tested walking pattern generators that can be used to generate omnidirectional motions with low computational requirements. These approaches employ a simple kinematics model (3D-LIPM) to reason about the balance of the robot. By using the model in a rotating environment we have found a simple way to understand and modify the rotational dynamics of the platform. In this way, we have presented a set of equations to reduce the angular momentum of the robot. The angular momentum

controller is an excellent complement to a walking pattern generator designed to prevent rotations.

There are, however, occasions when the angular momentum controller is not able to stabilize the robot. Specially if the motions requested to the locomotion system are not realistic. For that cases, another controller is necessary that adapts the duration of the steps to the state of the CoM. By using the 3D-LIPM we can predict the optimal exchange time for the support foot, and therefore the proper duration of the step, if the state of the CoM is known. Nevertheless, in realistic situations, the accurate estimation of the pose of the robot in a fast changing dynamic environment is highly complicated. To achieve robust adaption to the unexpected states of the robot dynamics, we have presented here a minimum effect timing controller. By adding minimal increments to the duration of the step in every control cycle we achieve robust performance with low accuracy requirements.

Chapter 6

Experimental Validation

6.1 Introduction

In this chapter we validate the robustness of the proposed balance control approach and study the improvements provided by the feedback controllers both in regular walking situations and under the action of disturbances.

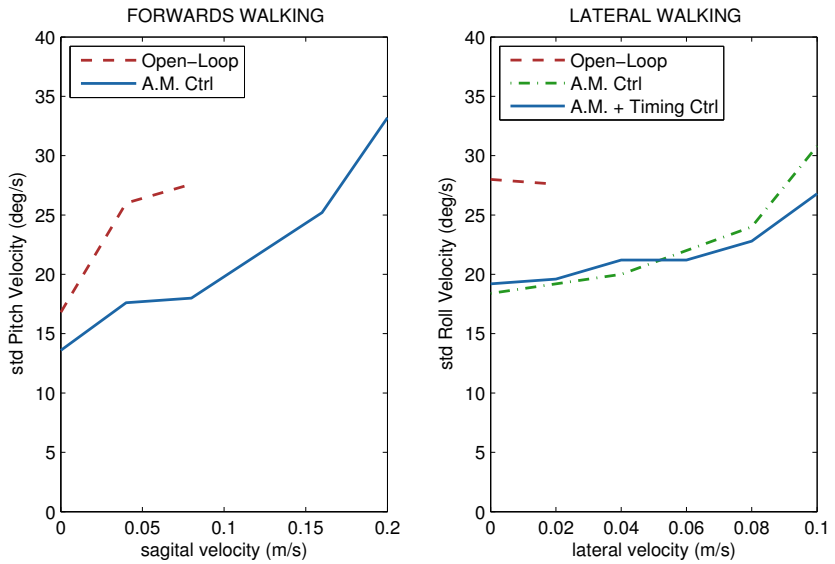
To analyze the effect of the controller on the stability of undisturbed walked, we will measure the stability of the gait for different lateral and sagittal velocities with the controllers enabled and disabled. As a measure of stability we have employed the standard deviation of the raw measures of the gyrometers. In this way we avoid the effect of the zero-offset drift of the gyrometers in the results.

The disturbance rejection performance of the controllers is studied for still, walking on spot and forward walking situations. The experiments study the performance of the controller in the sagittal and in the frontal plane separately. While the experiments that focus on the sagittal dimension allow us to measure more accurately the effect of the angular momentum controller, the experiments that study the performance on the frontal plane permit to observe specifically the effect of the timing controller.

6.2 Gait Stability

Figure 6.1 illustrates the stability performance of the robot with different configurations during forward and sidewards walking. The height of the CoM was fixed at an height of 260 mm and the scheduled duration for the step was 250 ms. Each configuration of the balance system was tested with step sizes ranging from 0 to 50 mm with increments of 10mm. In every test the measures provided by gyrometers were recorded for 35 s, and the first 5 s were discarded.

Figure 6.1a compares the stability performance during forward walking with



(a) (b)

Figure 6.1: Gait stability with different configurations of the balance system.

the angular momentum controller of the sagittal dimension enabled and disabled. The figure shows that the gait is more stable when the controller is enabled, and the robot is able to reach a velocity of 0.2 m/s. When the controller is disabled, on the other hand, the instability grows abruptly and the robot cannot go faster than 0.08 m/s.

For the sidewards walking, displayed in Figure 6.1b, we have compared three different configurations of the balance system. In addition to the angular momentum controller, we have measured the action of the timing controller. Again, the angular momentum controller substantially improves stability and allows a maximum speed of 0.1 m/s. Since it is not possible to cross the feet, the robot only can advance laterally with one of the feet, hence the speed attained is more reduced than in the forward walking experiment. When the angular momentum controller is disabled, the instability grows and the robot can barely move (0.02 m/s) stably. The timing controller, for his part, only improves slightly the stability for the top step sizes.

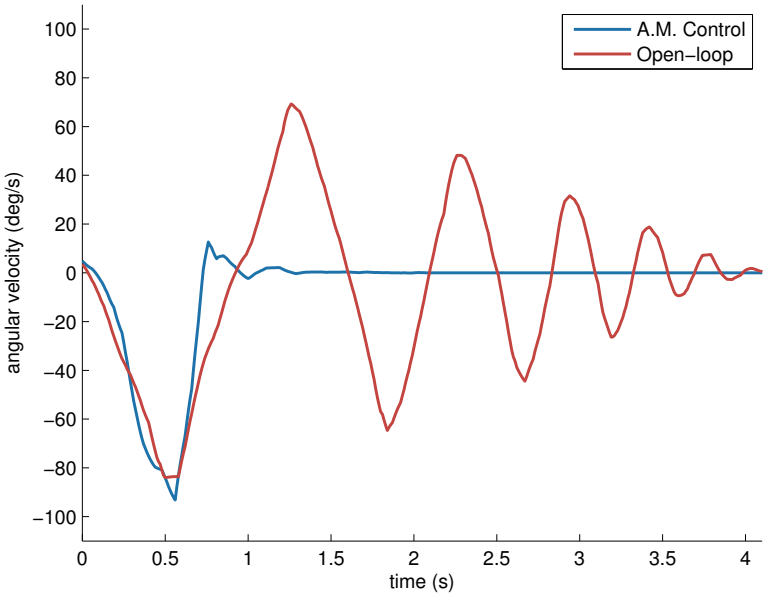


Figure 6.2: Lateral disturbance on still robot. The angular momentum controller reduces the stabilization time.

6.3 Disturbance Rejection

6.3.1 Still robot

The goal of this experiment is to validate the performance of the angular velocity controller in combination with the preview control approach. During this experiment, feet motion is disabled so that the robot stands still, but the linear predictive controller is active and the ZMP reference is held fixed in the middle between the two feet. With this setup,the robot is tilted laterally by 45 degrees, so that it is standing on the outer edge of the sole. Then, the robot is released and allowed to freely swing back towards the initial standing position. Fig. 6.2 illustrates the performance gained from the angular velocity controller. When the controller is disabled, the angular momentum accumulated by the robot during the time that the torso needs to recover the vertical position compels the robot to keep rotating, and thus the robot oscillates from one side to the other during the next four seconds. On the other hand, the angular velocity controller notably compensates the overshoot of the angular velocity and the equilibrium position is recovered in 1.25 seconds. When the angular velocity controller is disabled, the magnitude of the peak angular velocity in the first rebound is reduced only by 18%. Enabling the controller increases the reduction rate to 88%.

6.3.2 Walking on spot

6.3.2.1 Lateral Plane

In the undisturbed walking experiment we did not appreciate significant improvement in the performance due to the activation of the timing controller. To understand the advantages provided by this controller we must observe the performance of the robot under lateral disturbances.

To generate the lateral disturbance, the robot will be hit manually with a hollow plastic stick while it performs a zero velocity walking, or walking on the spot. Figure 6.3 depicts a sequence of frames corresponding to the film of the experiment. The sampling frequency of the film frames is 12 Hz.

Figure 6.4 shows the performance of the robot against a lateral disturbance at t=0 s. Both in the upper and the lower rows we show the trajectory of the CoM delivered to the inverse kinematics module and the estimated trajectory of the CoM obtained after rotating the reference frame with the sole angle estimated.

In the configuration where the timing control is disabled (upper row), the robot keeps the stepping frequency unaltered after the disturbance, which causes the loss of synchrony between the oscillation of the real position of the CoM and the oscillation generated by the model predictive controller. The desynchronization involves that for some steps the weight of the robot is supported by a foot that is scheduled as swing foot, posing a major threat to balance. In this example, the balance is not fully recovered until 150 ms later.

The activation of the timing controller avoids the loss of synchrony between real and scheduled CoM trajectories. In the lower row of Figure 6.4 the robot recovers the regular pace after 60 ms even though the magnitude of the lateral disturbance is considerably higher.

Figure 6.3: Lateral disturbance rejection.

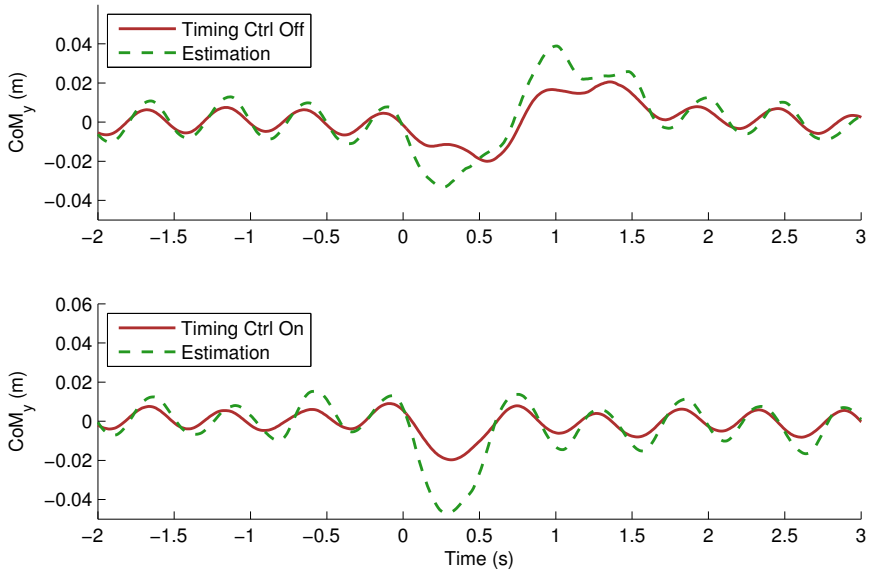


Figure 6.4: Performance of the timing controller against lateral disturbances.

6.3.2.2 Sagittal Plane

The second experiment for disturbance rejection while walking on the spot demonstrates the capacity of the balance control approach to reject disturbances in the sagittal plane. In this experiment, the robot is hit by a soccer ball that hangs as a pendulum from two 4m nylon cords, as illustrated in Figure 6.5. The mass of the ball is 450 g (equivalent to 10% of the mass of the robot) and its diameter is 200 mm. The impact point with the robot is located on its back, at an height of 290 mm, i.e., 30 mm over the CoM position. Before releasing the ball, it is moved towards one of the sides until it reaches the height of 430 mm over the impact point. The potential energy acquired is therefore 1.89 J. The robot receives the impact while it is walking on the spot with the target ZMP located on the vertical projections of the ankles over the soles.

The robot receives 10 impacts with the angular momentum controller enabled and 10 impacts with the controller disabled. Figure 6.6 displays the pitch angle estimated for the sole of the robot in the curse of the 20 impacts. The difference of performance is noticeable, since the robot falls backwards in all 10 experiences without the controller and manages to reject the disturbance in all cases when the controller is enabled.

The reason why the robot tips over backwards even though it receives the impact in its back is related to the asymmetrical layout of the humanoid feet, whose ankles are closer to the heel than to the toe, as was discussed in Section 5.3.

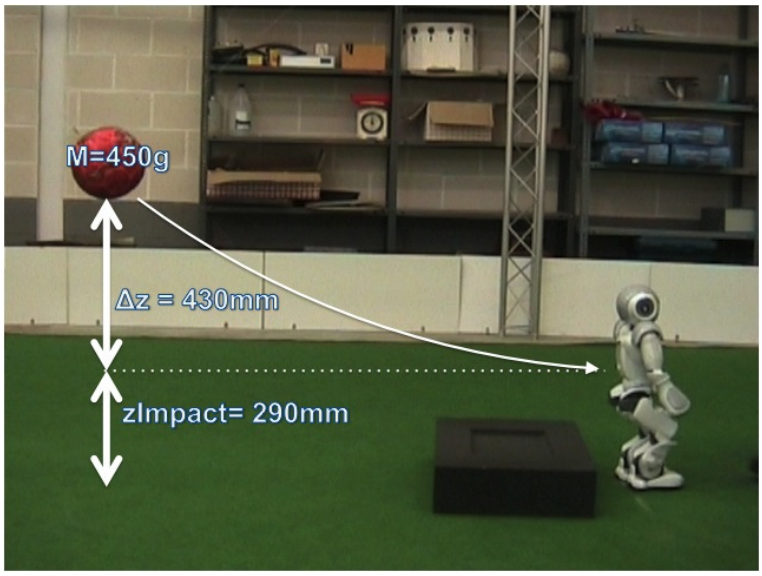


Figure 6.5: Disturbance rejection experiment layout.

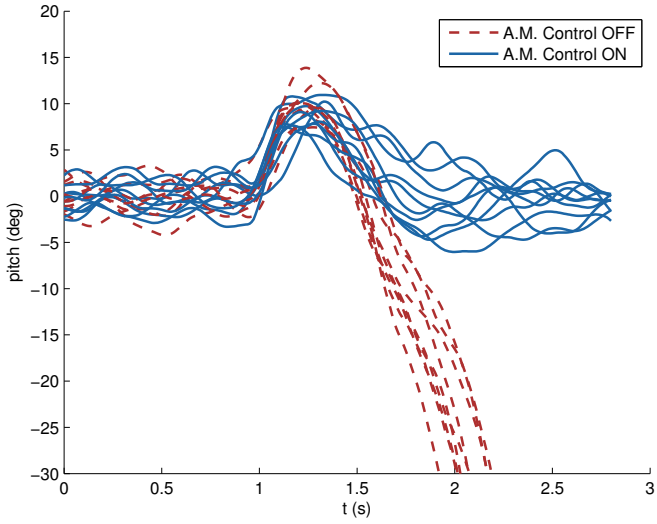


Figure 6.6: Disturbance rejection in the sagittal plane.

6.3.3 Forward motion

The last experiment concerned with disturbance rejection is devoted to illustrate the performance of the robot against disturbances during forward walking. The goal of this experiment is not to obtain any quantitative measure of the stability improvements generated by the controllers but to validate qualitatively their altogether performance.

Figure 6.7 displays the sequence of video frames corresponding to this experiment. Once again, the sampling frequency is 12 Hz. In the sequence of frames we can appreciate how the robot is hit in its back by an spheric body whose mass is 450 grams. The robot is walking forward with a velocity of 0.12 m/s when it receives the impact, which causes a perceptive bending of the body of the robot. Since the feet are not aligned at the impact instant, the forward bending also causes lateral instability. In this situation, the event of a step of the robot would result in a significant collision with the ground that would probably lead to a fall. Nevertheless, the activation of the timing controller delays the beginning of the next step until the rotation in the frontal plane is reduced by the angular momentum. Finally, in the bottom row of the figure, the robot has recovered balance and resumes the standard walking pattern for forward motion.

6.4 Discussion

The previous experiments have demonstrated that the balance controller proposed in this work provides significant improvement of the balance of the robot both against disturbances and in undisturbed circumstances. To the contrary, the walking pattern that has been employed to generate the trajectories for the CoM and the swing foot, has displayed considerably low performance in the open-loop configuration.

The reason why the walking pattern generator is not able to generate stable gaits even in the absence of disturbances is certainly related to the lack of accuracy of the kinematic model employed: the 3D-LIMP model involves substantial simplifications in the distribution of the masses along the body. Moreover the servomotors of the robot cannot track exactly the commanded trajectories. Since the tracking error of the joints of the legs typically occur either in the frontal or in the sagittal plane, the error cumulates through the kinematic chains of the legs and becomes a serious complication at the level of the torso.

Additionally, the approach employed for the walking pattern showed satisfactory results in similar test performed in the Webots simulator, which implements the Open Dynamics Engine to simulate the body dynamics of the Nao robot. The positive results in simulation imply that the cause for the instabilities

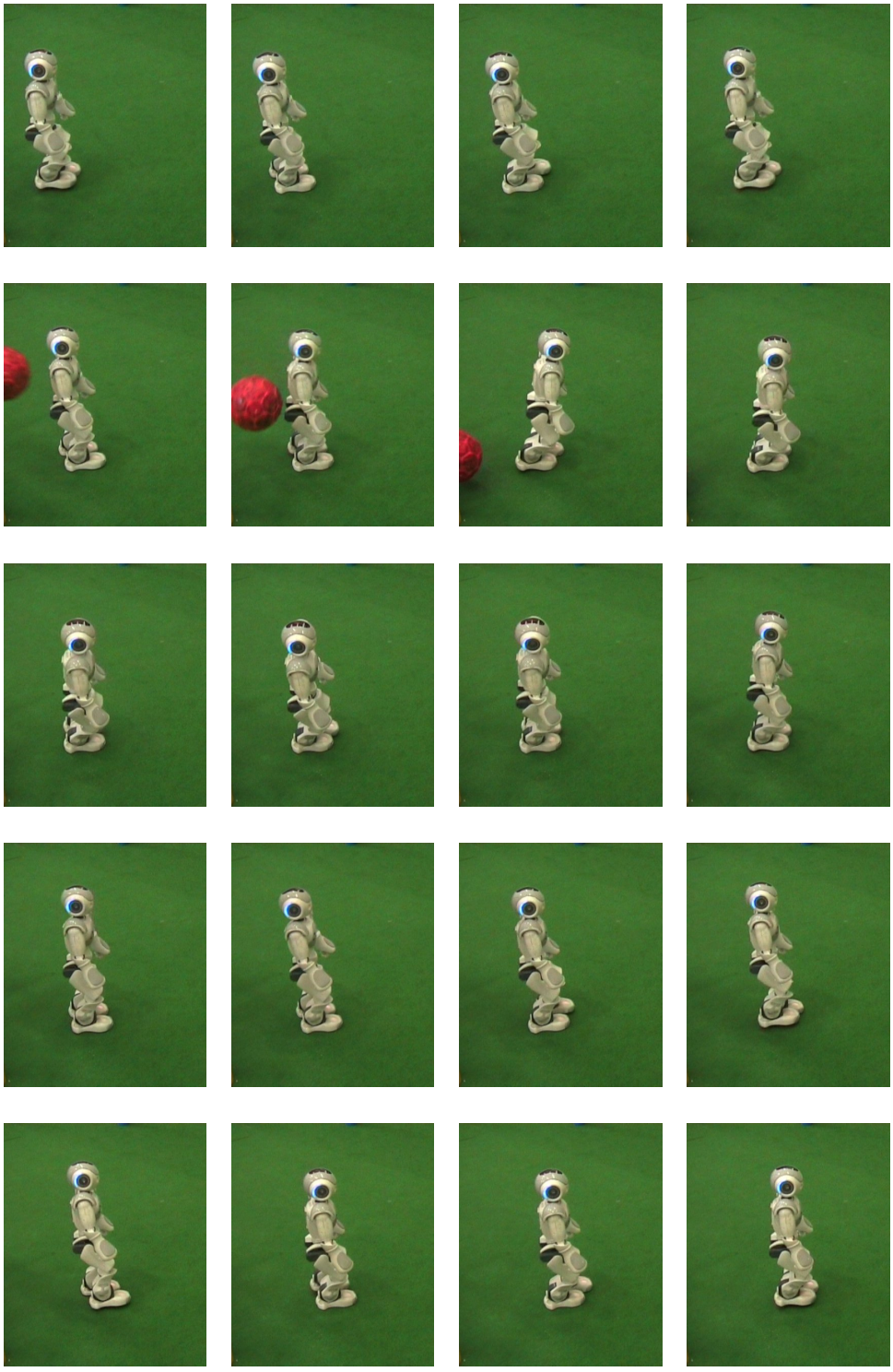


Figure 6.7: Sagittal disturbance rejection during forward walking.

must be related to the difference between the body model of the simulator and the real model of the robot.

The combination of the walking pattern generator with the balance controllers proposed in this work improves considerably the stability of the gait. The impact of the angular momentum controller in balance is notable even in the absence of disturbances. A common cause for the robot to tip over during regular walking is the cumulation of rotational inertia through consecutive steps. A possible cause for this is the relation between the natural frequency of the oscillations in the frontal or sagittal planes and certain stepping frequencies. However, the activation of the angular momentum controller effectively reduces the rotational velocity of the robot which prevents the amplification of rotational inertia between consecutive steps.

The reduction of the rotational velocity obtained by the angular momentum controller involves a drastic enhancement of the stability of the gait, both in the sagittal and in the frontal planes. The additional stability allows the robot to double its maximum forward velocity and to quintuplicate it in the lateral dimension.

Besides, the angular momentum controller has also demonstrated that it is a powerful tool to cope with external disturbances. When the robot is hit by an external body, the body applies a certain force at the impact point. This force faces a reaction force at the contact point between the sole of the robot and the ground. The result of the combination of this two forces is a torque around the contact point on the ground that causes the rotation of the robot. If the magnitude of the impact is high enough to cause the tipping over of the robot, the effect of the angular momentum controller will not be noticeable. However, if the impact only destabilizes the robot and does not cause its fall, the angular momentum controller will reduce the rotational velocity and prevent the robot to fall in subsquent steps.

Fundamental as it is the use of the angular momentum controller to improve balance, it requires a complementary controller to modify the duration of the steps in anomalous situations. For instance, if the robot is hit laterally, it will bend and it will need a minimum time to recover the vertical position. If the step finishes before the robot has recovered the vertical orientation, it will use the foot that supports the weight of the body to advance, which would probably lead to a tipping over. The timing controller that we propose monitors the feasibility of the planned trajectory for the CoM and adjusts the duration of the step in case the planed duration is not enough.

In summary, the experiments have demonstrated that the combined use of angular momentum and timing controllers improves notably the stability of the gait and allows the robot to perform stable motions and recover from impacts.

Chapter 7

Conclusion

In this final Chapter we summarize the main contributions of the thesis and outline the future works that could sprout from this research.

7.1 What has been achieved

Over the previous chapters, we have described a complete walking engine for a specific platform: the humanoid robot Nao. To achieve the development of a robust gait for the robot, several challenges had to be addressed. Nevertheless, the focus of this work is placed in the control of the balance of the robot. Considering the Nao robot has been envisaged as a widely affordable platform, the limited performance of its components frequently offers a challenge in the control area. Hence, robust feedback strategies are required to ensure that the controllers make the most of the available information.

In this way, a set of controllers has been presented that effectively improves the balance of a walking engine based on the preview control of the ZMP. One of the controller aims at reducing the rotational velocity of the platform, which has an positive effect in the stability of the gait. The second controller monitors the state of the system and supervises the feasibility of the commanded motions.

Although we have only shown the integration of the controllers in a walking engine which is based on the preview control of the ZMP, the simple control actions of both controllers would allow us to adapt them to different walking pattern generators. Therefore, simplicity and versatility are doubtlessly significant advantages of our approach.

In summary, the main achievements of this work are:

- Analysis of the rotational dynamics of the robot. We have derived from the inverted pendulum model a set of equations to describe the effect of the

manipulation of the CoM on the rotational velocity of the body.

- Design of an angular momentum controller to reduce rotational velocity around the contacts points between the robot and the ground. The controller is inferred from the study of the rotational dynamics of the robot and is based on the acceleration of the CoM to reduce the rotational velocity of the body.

- Design of timing controller to adapt footstep duration in real time. This controller monitors the dynamics of the CoM and the target trajectories for the ZMP. The controller only actuates in case the target trajectories for the ZMP cannot be tracked accurately and involve a risk for the stability of the gait. To estimate the timing requirements of the gait reference orbital levels are employed.

Even thought these are the main achievements, in order to obtain a robust gait, other challenges had to be faced. In this way, we can highlight several secondary achievements:

- Detailed analysis of the platform sensors and servomotors. The study of the accuracy of the sensors allowed us to determine the kind of information that can be obtained from them. For example, in the case of the FSRs we concluded that only contact information would be employed and not the exact location of the center of pressure. These decisions condition the information available for the feedback controllers, and are therefore paramount in the design of the balance system. Similarly, the evaluation of the sensorimotor performance produced estimations of the sensorimotor lag and backlash which are essential to the configuration of the balance controllers.

- Design of a Kalman filter to estimate the orientation of the body. The filter fuses measurements of accelerometers and gyrometers of the inertial unit and the contact information provided by the FSRs to estimate the orientation of the torso. Subsequently, through the use forwards kinematics operations, we obtain the pose of the feet of the robot in the ground frame.

- Development of inverse kinematics solution for Nao. The innovative configuration of the legs of Nao, with two coupled yaw-pitch joints, required the use of a new inverse kinematics approach. Our solution consists of three stages. The first two stages use P-controllers to prepare the inputs for the third stage, which is solved analytically. The resulting approach is both computationally efficient and accurate.

- Walking pattern generator design as an integration of existing approaches, to create the base trajectories for the omnidirectional gait.

Several experiments have tested the robustness of the balance control in a range of situations. The results obtained prove that the controllers improve the stability of the gait and allow the robot to reach higher velocities. Moreover, the additional stability allows the robot to reject external disturbances while performing regular motions.

7.2 Dissemination of the results

A number of papers concerning developments related to this thesis have been presented in national and international conferences and journals. Next, we provide a chronologically-sorted list of them:

- J. J. Alcaraz-Jiménez, D. Herrero-Pérez, and H. Martínez-Barberá, Robust Feedback Control of ZMP-Based Gait for Humanoid Robot Nao, to appear in International Journal of Robotics Research, 2013.

- J. J. Alcaraz-Jiménez, D. Herrero-Pérez, and H. Martínez-Barberá, A Simple Feedback Controller to Reduce Angular Momentum in ZMP-Based Gaits, International Journal of Advanced Robotic Systems, vol. 10, no. 102, 2012.

- J. J. Alcaraz-Jiménez, M. Missura, H. Martínez-Barberá, and S. Behnke, Lateral Disturbance Rejection for the Nao Robot, Lecture Notes in Computer Science. RoboCup 2012: Robot Soccer World Cup XVI, vol. 7500, pp. 1–12, 2013.

- F. Blanes, M. Muoz, J. Simó, H. Martínez-Barber, and J. J. Alcaraz-Jimnez, Arquitecturas de control sobre robots Nao en la SPL de Robocup, Robot, Nov. 2011.

- J. J. Alcaraz-Jiménez, D. Herrero-Pérez, and H. Martínez-Barberá, Omnidirectional Locomotion for the Hidalgos Humanoid Soccer Team, XII Workshop of Physical Agents, 2011.

- J. J. Alcaraz-Jiménez, D. Herrero-Pérez, and H. Martínez-Barberá, Motion planning for omnidirectional dynamic gait in humanoid soccer robots, Journal of Physical Agents, vol. 5, no. 1, pp. 25–34, 2011.

- J. J. Alcaraz-Jiménez, D. Herrero-Pérez, and H. Martínez-Barberá, A Closed-Loop Dribbling Gait for the Standard Platform League, Workshop on Humanoid Soccer Robots of the IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids), 2011.

- J. J. Alcaraz-Jiménez, D. Herrero-Pérez, and H. Martínez-Barberá, Planning CoM Trajectories for an Omnidirectional Dynamic Gait in Humanoid Soccer Robots, XI Workshop of Physical Agents, 2010.

- J. J. Alcaraz-Jiménez, D. Herrero-Pérez, and H. Martínez-Barberá, Control por comportamientos embebidos en robots personales, : IV Workshop Robocity 2030, Robots Personales y Asistenciales, 2008.

It is worth mentioning that the paper *Lateral Disturbance Rejection for the Nao Robot*, obtained the Best Paper Award in the Robocup Symposium 2012. Additionally, the developments performed for this thesis contributed to the victory of the "Hidalgos Team" in the RoboCup Mediterranean Open 2010.

7.3 Future works

In this thesis we have presented a set of controllers for the balance of the humanoid robot Nao. Although a high degree of robustness and stability has been attained, there remains a number of ways to achieve even better performance. The most evident approach is the addition of specific handcrafted functions to the signals of the walking pattern generator to create a more stable open-loop gait.

In this work, however, we have been concerned with the versatility of our approaches. Although we have performed all our experiments in Nao robots, the simple formulation of the controllers and the limited use of the information delivered by the sensors, provides evidences that the control approach can be easily extended to other platforms.

Additionally, the natural extension of the set of controllers proposed here is the combination with controllers for the step size and the pose of the swing foot. Similarly to the timing controller, this type of controllers monitor the feasibility of the commanded target position for the swing foot and adapt it in case of balance emergency. On the other hand, the use of the torso orientation to manipulate the angular momentum of the robot could also been explored, as a complement to the angular momentum controller based on the acceleration of the CoM.

References

[1] M. Abdallah and A. Goswami. A biomechanically motivated two-phase strategy for biped upright balance control. *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 1996–2001, 2005. 38

[2] K. Ahn and Y. Oh. Walking control of a humanoid robot via explicit and stable CoM manipulation with the angular momentum resolution. *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 2478–2483, 2006. 35

[3] Hrp-4 specifications. http://www.aist.go.jp/aist_j/press_release/pr2010/pr20100915/pr20100915.html. Accessed: 23/10/2012. 10

[4] M. Arbulú, D. Kaynov, L. Cabas, and C. Balaguer. The Rh-1 full-size humanoid robot: Design, walking pattern generation and control. *Applied Bionics and Biomechanics*, 6(3-4):301–344, 2009. 11

[5] C. Azevedo, P. Poignet, and B. Espiau. Artificial locomotion control: from human to robots. *Robotics and Autonomous Systems*, 47(4):203–223, 2004. 38

[6] C. E. Bauby and A. D. Kuo. Active control of lateral balance in human walking. *Journal of biomechanics*, 33(11):1433–1440, 2000. 40

[7] S. Behnke. Online trajectory generation for omnidirectional biped walking. *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 1597–1603, 2006. 25

[8] G. Bishop and G. Welch. An introduction to the kalman filter. *Proc of SIGGRAPH, Course*, 8:27599–23175, 2001. 41, 42

[9] D. J. Block, K. J. Åström, and M. W. Spong. The Reaction Wheel Pendulum. *Synthesis Lectures on Controls and Mechatronics*, 1(1):1–105, Jan. 2007. 19

[10] S. R. Buss. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. *University of California, San Diego, Typeset manuscript, available from http://math. ucsd. edu/ sbuss/ResearchWeb*, 2004. 61

[11] R. Chalodhorn, D. B. Grimes, K. Grochow, and R. P. N. Rao. Learning to walk through imitation. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2007. 25

[12] B.-K. Cho, S.-S. Park, and J.-h. Oh. Stabilization of a hopping humanoid robot for a push. In *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, pages 60–65, 2010. 39

[13] Y. Choi, D. Kim, and B.-J. You. On the walking control for humanoid robot based on the kinematic resolution of CoM Jacobian with embedded motion. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, 2006. 35

[14] S. Coros, P. Beaudoin, and M. van de Panne. Generalized biped walking control. *ACM Transactions on Graphics (TOG)*, 29(4):130, 2010. 39

[15] S. Czarnetzki, S. Kerner, and O. Urbann. Observer-based dynamic walking control for biped robots. *Robotics and Autonomous Systems*, 57(8):839–845, July 2009. 37

[16] S. Czarnetzki, S. Kerner, and O. Urbann. Applying dynamic walking control for biped robots. *RoboCup 2009: robot soccer world cup XIII*, pages 69–80, 2010. 37

[17] D. Dimitrov, P. B. Wieber, H. J. Ferreau, and M. Diehl. On the implementation of model predictive control for on-line walking pattern generation. *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 2685–2690, 2008. 32

[18] D. Dimitrov, P. B. Wieber, O. Stasse, H. J. Ferreau, and H. Diedam. An optimized linear model predictive control solver for online walking motion generation. *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 1171–1176, 2009. 32

[19] F. Faber and S. Behnke. Stochastic optimization of bipedal walking using gyro feedback and phase resetting. *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*, pages 203–209, 2007. 36

[20] Hoap-3 specifications. http://home.comcast.net/~jtechsc/HOAP-3_Spec_Sheet.pdf. Accessed: 23/10/2012. 11

[21] E. Garcia, J. Estremera, and P. G. de Santos. A classification of stability margins for walking robots. *Proc. of the 2002 Int. Symp. on Climbing and Walking Robots*, 2002. 20

[22] M. Garcia, A. Chatterjee, A. Ruina, and M. Coleman. The simplest walking model: Stability, complexity, and scaling. *J BIOMECH ENG TRANS ASME*, 120(2):281–288, 1998. 33

[23] A. Goswami. Foot rotation indicator (FRI) point: a new gait planning tool to evaluate postural stability of biped robots. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, pages 47–52, 1999. 21, 22

[24] A. Goswami and V. Kallem. Rate of change of angular momentum and balance maintenance of biped robots. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, pages 3785–3790, 2004. 35

[25] D. Gouaillier, C. Collette, and C. H. R. H. . t. I.-R. I. C. o. Kilner. Omni-directional closed-loop walk for NAO. In *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, Nov. 2010. 37

[26] D. Gouaillier, V. Hugel, P. Blazevic, C. Kilner, J. Monceaux, P. Lafourcade, B. Marnier, J. Serre, and B. Maisonnier. The NAO humanoid: a combination of performance and affordability. *arXiv.org*, cs.RO, July 2008. 11

[27] D. Gouaillier, V. Hugel, P. Blazevic, C. Kilner, J. Monceaux, P. Lafourcade, B. Marnier, J. Serre, and B. Maisonnier. Mechatronic design of NAO humanoid. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 769–774, 2009.

[28] C. Graf, A. Härtl, T. Röfer, and T. Laue. A robust closed-loop gait for the standard platform league humanoid. *Proceedings of the Fourth Workshop on Humanoid Soccer Robots in conjunction with the*, pages 30–37, 2009. 26

[29] C. Graf and T. Röfer. A closed-loop 3D-LIPM gait for the RoboCup Standard Platform League humanoid. *Proceedings of the Fourth Workshop on Humanoid Soccer Robots in conjunction with the*, 2010. 37

[30] C. Graf and T. Röfer. A center of mass observing 3D-LIPM gait for the RoboCup Standard Platform League humanoid. *RoboCup 2011: Robot Soccer World Cup XV*, pages 102–113, 2012. 37

[31] Grishin robotics market overview. http://grishinrobotics.com/. Accessed: 23/10/2012. 8

[32] K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popović. Style-based inverse kinematics. *ACM Transactions on Graphics (TOG)*, 23(3):522–531, 2004. 61

[33] I. Ha, Y. Tamura, H. Asama, J. Han, and D. W. Hong. Development of open humanoid platform DARwIn-OP. In *SICE Annual Conference (SICE), 2011 Proceedings of*, pages 2178–2181, 2011. 11

[34] K. Harada, S. Kajita, K. Kaneko, and H. Hirukawa. ZMP analysis for arm/leg coordination. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, pages 75–81, 2003. 22

[35] K. Harada, S. Kajita, K. Kaneko, and H. Hirukawa. An analytical method on real-time gait planning for a humanoid robot. *Humanoid Robots, 2004 4th IEEE/RAS International Conference on*, 2:640–655, 2004. 28

[36] K. Hirai. Current and future perspective of Honda humamoid robot. *Intelligent Robots and Systems, 1997. IROS'97., Proceedings of the 1997 IEEE/RSJ International Conference on*, 2:500–508, 1997. 27

[37] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka. The development of Honda humanoid robot. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, pages 1321–1326, 1998. 36, 38

[38] H. Hirukawa, S. Hattori, K. Harada, S. Kajita, K. Kaneko, F. Kanehiro, K. Fujiwara, M. R. Morisawa, and A. . I. . P. . I. I. C. on. A universal stability criterion of the foot contact of legged robots - adios ZMP. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, Dec. 2006. 23

[39] A. Hofmann, M. Popovic, and H. Herr. Exploiting angular momentum to enhance bipedal center-of-mass control. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 4423–4429, 2009. 35

[40] Asimo 2011 specifications. http://world.honda.com/ASIMO/technology/2011/specification/. Accessed: 23/10/2012. 10

[41] Q. Huang, K. Kaneko, K. Yokoi, S. Kajita, T. Kotoku, N. Koyachi, H. Arai, N. Imamura, K. Komoriya, and K. Tanie. Balance control of a piped robot combining off-line pattern with real-time modification. *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, 4:3346–3352, 2000. 39

[42] International federation of robotics. http://www.ifr.org/service-robots/. Accessed: 23/10/2012. 7

[43] A. J. Ijspeert. 2008 Special Issue: Central pattern generators for locomotion control in animals and robots: A review. *Neural Networks*, 21(4):642–653, 2008. 25

[44] S. Kagami, T. Kitagawa, K. Nishiwaki, T. Sugihara, M. Inaba, and H. Inoue. A fast dynamically equilibrated walking trajectory generation method of humanoid robot. *Autonomous Robots*, 12(1):71–82, 2002. 28

[45] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, 2:1620–1626, 2003. 30

[46] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Resolved momentum control: Humanoid motion planning based on the linear and angular momentum. *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, 2:1644–1650, 2003. 34

[47] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Yokoi, and H. Hirukawa. A realtime pattern generator for biped walking. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, pages 31–37, 2002. 28

[48] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa. The 3D Linear Inverted Pendulum Mode: A simple modeling for a biped walking pattern generation. *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, 1:239–246, 2001. 28

[49] S. Kajita, M. Morisawa, K. Harada, K. Kaneko, F. Kanehiro, K. Fujiwara, and H. Hirukawa. Biped walking pattern generator allowing auxiliary zmp control. *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 2993–2999, 2006. 38

[50] S. Kajita, T. Nagasaki, K. Kaneko, and H. Hirukawa. ZMP-Based Biped Running Control. *Robotics & Automation Magazine, IEEE*, 14(2):63–72, 2007. 38

[51] S. Kajita, K. Tani, and A. Kobayashi. Dynamic walk control of a biped robot along the potential energy conserving orbit. In *Intelligent Robots and Systems '90. 'Towards a New Frontier of Applications', Proceedings. IROS '90. IEEE International Workshop on*, pages 789–794, 1990.

[52] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960. 41

[53] D. Kim, Y. Choi, and C. Kim. Motion-embedded cog jacobian for a real-time humanoid motion generation. *2nd International Conference on Informatics in Control, Automation and Robotics (ICINCO 2005)*, pages 55–61, 2005. 35

[54] J. Y. Kim, I. W. Park, and J. H. Oh. Walking control algorithm of biped humanoid robot on uneven and inclined floor. *Journal of Intelligent & Robotic Systems*, 48(4):457–484, 2007. 36

[55] T. Komura, H. Leung, S. Kudoh, and J. Kuffner. A feedback controller for biped humanoids that can counteract large perturbations during gait. *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 1989–1995, 2005. 17

[56] T. Komura, A. Nagano, H. Leung, and Y. Shinagawa. Simulating pathological gait using the enhanced linear inverted pendulum model. *Biomedical Engineering, IEEE Transactions on*, 52(9):1502–1513, 2005. 17

[57] S. Kudoh and T. Komura. C2 continuous gait-pattern generation for biped robots. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, pages 1135–1140, 2003. 17

[58] S. Kuindersma, R. Grupen, and A. Barto. Learning dynamic arm motions for postural recovery. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pages 7–12, 2011. 37

[59] J. Kulk and J. Welsh. A low power walk for the NAO robot. *Proceedings of the 2008 Australasian Conference on Robotics & Automation (ACRA-2008), J. Kim and R. Mahony, Eds*, pages 1–7, 2008. 55

[60] A. D. Kuo. Stabilization of lateral motion in passive dynamic walking. *The International Journal of Robotics Research*, 18(9):917–930, 1999. 33

[61] A. D. Kuo. Energetics of actively powered locomotion using the simplest walking model. *J BIOMECH ENG TRANS ASME*, 124(1):113–120, 2002. 33

[62] A. D. Kuo, J. M. Donelan, and A. Ruina. Energetic consequences of walking like an inverted pendulum: step-to-step transitions. *Exercise and sport sciences reviews*, 33(2):88–97, 2005. 33

[63] S. H. Lee and A. Goswami. Reaction mass pendulum (RMP): An explicit model for centroidal angular momentum of humanoid robots. *Robotics and Automation, 2007 IEEE International Conference on*, pages 4667–4672, 2007. 19

[64] S. H. Lee and A. Goswami. A momentum-based balance controller for humanoid robots on non-level and non-stationary ground. *Autonomous Robots*, pages 1–16, 2012. 35

[65] H.-o. Lim and A. Takanishi. Realization of continuous biped walking. In *Systems, Man, and Cybernetics, 2001 IEEE International Conference on*, Dec. 2001. 27

[66] A. Macchietto, V. Zordan, and C. R. Shelton. Momentum control for balance. *ACM Transactions on Graphics (TOG)*, 28(3):1–8, 2009. 35

[67] T. McGeer. Stability and control of two-dimensional biped walking. *Center for Systems Science, Simon Fraser University, Burnaby, BC, Canada, Tech. Rep*, 1, 1988. 33

[68] T. McGeer. Passive dynamic walking. *The International Journal of Robotics Research*, 9(2):62–82, 1990. 33

[69] Ç. Meriçli and M. Veloso. Biped Walk Learning On Nao Through Playback and Real-time Corrective Demonstration. *In Workshop on Agents Learning Interactively from Human Teachers, 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, 2010. 26

[70] M. Missura and S. Behnke. Lateral capture steps for bipedal walking. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pages 401–408, 2011. 39

[71] M. Mistry, J. Nakanishi, G. Cheng, and S. Schaal. Inverse kinematics with floating base and constraints for full body humanoid robot control. *Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on*, pages 22–27, 2008. 61

[72] M. Morisawa, K. Harada, S. Kajita, K. Kaneko, F. Kanehiro, K. Fujiwara, S. Nakaoka, and H. Hirukawa. A Biped Pattern Generation Allowing Immediate Modification of Foot Placement in Real-time. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 581–586, 2006. 39

[73] M. Morisawa, F. Kanehiro, K. Kaneko, S. Kajita, and K. Yokoi. Reactive biped walking control for a collision of a swinging foot on uneven terrain. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pages 768–773, 2011. 37, 39

[74] M. Morisawa, F. Kanehiro, K. Kaneko, N. Mansard, J. Sola, E. Yoshida, K. Yokoi, and J. P. Laumond. Combining suppression of the disturbance and reactive stepping for recovering balance. *Proc. on IEEE Int. Conf. on Intelligent Robots and Systems*, pages 3150–3156, 2010. 39

[75] J. Müller, T. Laue, and T. Röfer. Kicking a ball–modeling complex dynamic motions for humanoid robots. *RoboCup 2010: Robot Soccer World Cup XIV*, pages 109–120, 2011. 37

[76] K. Nagasaka, H. Inoue, and M. Inaba. Dynamic walking pattern generation for a humanoid robot based on optimal gradient method. In *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on*, Dec. 1999. 27

[77] K. Nishiwaki and S. Kagami. Online walking control system for humanoids with short cycle pattern generation. *The International Journal of Robotics Research*, 28(6):729–742, 2009. 32

[78] K. Nishiwaki, J. Kuffner, S. Kagami, M. Inaba, and H. Inoue. The experimental humanoid robot H7: a research platform for autonomous behaviour. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1850):79–107, 2007. 37

[79] R. D. Olfati-Saber and C. . P. of the 40th IEEE Conference on. Global stabilization of a flat underactuated system: the inertia wheel pendulum. In *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, Dec. 2001. 19

[80] Y. C. Pai and J. Patton. Center of mass velocity-position predictions for balance control. *Journal of biomechanics*, 30(4):347–354, 1997. 20

[81] Reem-b specifications. http://www.pal-robotics.com/robots/reem-b/33-reem-b-spec. Accessed: 23/10/2012. 10

[82] J. H. Park and H. C. Cho. An online trajectory modifier for the base link of biped robots to enhance locomotion stability. *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, 4:3353–3358, 2000. 17, 38

[83] J. H. Park and K. D. Kim. Biped robot walking using gravity-compensated inverted pendulum mode and computed torque control. *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, 4:3528–3533, 1998. 17

[84] J. Peters, M. Mistry, F. Udwadia, J. Nakanishi, and S. Schaal. A unifying framework for robot control with redundant DOFs. *Autonomous Robots*, 24(1):1–12, 2008. 61

[85] M. Popovic, A. Hofmann, and H. Herr. Angular momentum regulation during human walking: biomechanics and control. *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, 3:2405–2411, 2004. 34

[86] J. Pratt, J. Carff, S. Drakunov, and A. Goswami. Capture Point: A Step toward Humanoid Push Recovery. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 200–207, 2006. 17, 19

[87] J. Pratt, C. M. Chew, A. Torres, P. Dilworth, and G. Pratt. Virtual model control: An intuitive approach for bipedal locomotion. *The International Journal of Robotics Research*, 20(2):129–143, 2001. 26

[88] M. Raibert, K. Blankespoor, G. Nelson, and R. Playter. Bigdog, the rough–terrain quadruped robot. *Proceedings of the 17th World Congress*, pages 10823–10825, 2008. 38

[89] S. A. Setiawan, J. Yamaguchi, S.-H. Hyon, and A. Takanishi. Physical interaction between human and a bipedal humanoid robot-realization of human-follow walking. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, 1999. 27

[90] C. L. Shih. The dynamics and control of a biped walking robot with seven degrees of freedom. *Journal of dynamic systems, measurement, and control*, 118(4):683–690, 1996. 20

[91] Qrio specifications. http://www.sonyaibo.net/aboutqrio.htm. Accessed: 23/10/2012. 10

[92] M. W. Spong, P. Corke, and R. Lozano. Nonlinear control of the reaction wheel pendulum. *Automatica*, 37(11):1845–1851, 2001. 19

[93] J. Strom, G. Slavov, and E. Chown. Omnidirectional walking using ZMP and preview control for the nao humanoid robot. *RoboCup 2009: robot soccer world cup XIII*, pages 378–389, 2010. 32

[94] T. Sugihara and Y. Nakamura. Whole-body cooperative balancing of humanoid robot using COG Jacobian. *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, 3:2575–2580, 2002. 36

[95] T. Sugihara, Y. Nakamura, and H. Inoue. Real-time humanoid motion generation through ZMP manipulation based on inverted pendulum control. *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, 2:1404–1409, 2002. 28

[96] A. Takanishi, H. Lim, M. Tsuda, and I. Kato. Realization of dynamic biped walking stabilized by trunk motion on a sagittally uneven surface. *Intelligent Robots and Systems' 90.'Towards a New Frontier of Applications', Proceedings. IROS'90. IEEE International Workshop on*, pages 323–330, 1990. 27

[97] A. Takanishi, T. Takeya, H. Karaki, and I. Kato. A control method for dynamic biped walking under unknown external force. *Intelligent Robots and Systems' 90.'Towards a New Frontier of Applications', Proceedings. IROS'90. IEEE International Workshop on*, pages 795–801, 1990. 36

[98] T. Takenaka. The control system for the Honda humanoid robot. *Age and Ageing*, 35(Supplement 2):ii24–ii26, Sept. 2006. 38

[99] T. Takenaka, T. Matsumoto, T. Yoshiike, T. Hasegawa, S. Shirokura, H. Kaneko, and A. Orita. Real time motion generation and control for biped robot -4th report: Integrated balance control-. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 1601–1608, 2009. 36, 38

[100] D. Tolani, A. Goswami, and N. I. Badler. Real-time inverse kinematics techniques for anthropomorphic limbs. *Graphical models*, 62(5):353–388, 2000. 62

[101] J. Urata, K. Nshiwaki, Y. Nakanishi, K. Okada, S. Kagami, and M. Inaba. Online decision of foot placement using singular lq preview regulation. *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pages 13–18, 2011. 39

[102] M. Vukobratovic and B. Borovac. Zero-moment point—thirty five years of its life. *International Journal of Humanoid Robotics*, 1(01):157–173, 2004. 21

[103] M. Vukobratovic, A. A. Frank, and D. Juricic. On the Stability of Biped Locomotion. *Biomedical Engineering, IEEE Transactions on*, (1):25–36, 1970. 21

[104] P. B. Wieber. On the stability of walking systems. *Proceedings of the international workshop on humanoid and human friendly robotics*, 2002. 21

[105] P. B. Wieber. Trajectory free linear model predictive control for stable walking in the presence of strong perturbations. *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 137–142, 2006. 32, 84, 86

[106] M. Wisse. Essentials of dynamic walking; Analysis and design of two-legged robots. *PhD. Thesis*, 2004. 33

[107] F. Xue, X. Chen, J. Liu, and D. Nardi. Real time biped walking gait pattern generator for a real robot. *RoboCup 2011: Robot Soccer World Cup XV*, pages 210–221, 2012. 28

[108] J. Yamaguchi, S. Inoue, D. Nishino, and A. Takanishi. Development of a bipedal humanoid robot having antagonistic driven joints and three DOF trunk. In *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, 1998. 27

[109] S. J. Yi, B. T. Zhang, D. Hong, and D. D. Lee. Learning full body push recovery control for small humanoid robots. *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2047–2052, 2011. 38

[110] S. J. Yi, B. T. Zhang, D. Hong, and D. D. Lee. Online learning of a full body push recovery controller for omnidirectional walking. *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pages 1–6, 2011. 38

[111] S. J. Yi, B. T. Zhang, D. Hong, and D. D. Lee. Practical bipedal walking control on uneven terrain using surface learning and push recovery. *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3963–3968, 2011. 38

[112] S. Yun and A. Goswami. Momentum-based reactive stepping controller on level and non-level ground for humanoid robot push recovery. *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3943–3950, 2011. 39