The 10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI 2013)
Oct. 30 – Nov. 2, 2013 at Ramada Plaza, Jeju, Korea

Hybrid gait pattern generator capable of rapid and dynamically consistent pattern regeneration

Przemyslaw Kryczka[1], Petar Kormushev[2], Kenji Hashimoto[3], Hun-Ok Lim[4],
Nikos G. Tsagarakis[5], Darwin G. Caldwell[6] and Atsuo Takanishi[7]

[1]Graduate School of Advanced Science and Engineering, Waseda University, Tokyo, Japan
(contact@takanishi.mech.waseda.ac.jp)
[2,6,7]Department of Advanced Robotics, Istituto Italiano di Tecnologia, via Morego 30, 16163 Genova, Italy
[3]Faculty of Science and Engineering, Waseda University, Tokyo, Japan
[4]Faculty of Engineering, Kanagawa University, Kanagawa, Japan
[4,5]Department of Modern Mechanical Engineering, Waseda University, Tokyo, Japan

Abstract - We propose a two-stage gait pattern generation scheme for the full-scale humanoid robots, that considers the dynamics of the system throughout the process. The fist stage is responsible for generating the preliminary motion reference, such as step position, timing and trajectory of Center of Mass (CoM), while the second stage serves as dynamics filter and modifies the initial references to make the pattern stable on the full-scale multi-degree-of-freedom humanoid robot. The approach allows employment of easy to use models for motion generation, yet the use of the dynamics filtering ensures that the pattern is safe to execute on the real-world humanoid robot. The paper contains description of two approaches used in the first and second stage, as well as experimental results proving the effectiveness of the method. The fast calculation time and the use of the system's dynamic state as initial conditions for pattern generation makes it a good candidate for the real-time gait pattern generator.

Keywords - bipedal locomotion, humanoid robots, real-time control, pattern generation, LIPM, preview control

1. Introduction

The control of humanoid robots, which are under-actuated and highly redundant structures is a very complicated task. In most of the locomotion control related methodologies, applied to the full-scale humanoid robots, the whole robot is simplified either to the single mass [1], [2] or to the set of masses [3]. Most of the methods eventually results in the gait pattern, which is the set of reference trajectories that completely define the spatial configuration of the robot in time.

For the humanoid robot to be useful for the society it has to be able to easily navigate in the human daily-life environment, which on one hand comprises a variety of surfaces that are difficult to model, while on the other hand is very dynamic and full of moving objects. To be able to handle this kind of environments the robot has to be equipped with the real-time control algorithms that are able to very quickly respond to the dynamic changes in the environment or to the changes in the motion that result from the inaccuracies in the models.

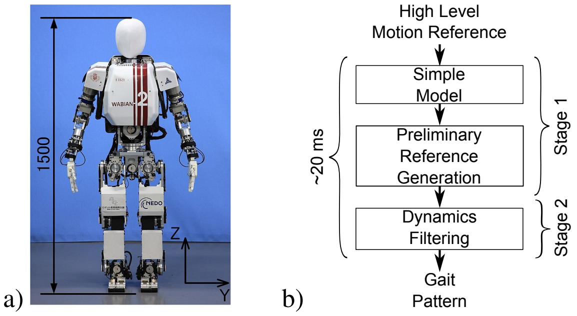

Fig. 1 a) Our research platform WABIAN-2R; b) The outline of the proposed two-stage pattern generation scheme.

In our research we focus on the model based pattern generation. There are two major approaches to the model based gait pattern generation. The first approach is based on the single-mass models while the second is based on the multibody dynamics. The former one pays strong attention to the natural dynamics of the locomotion and uses simple models, often being an inverted pendulum with extension of the foot to account for the ground contact or the rotating mass to account for the upper body inertia [2], [4], [5]. However, since they do not include the full body dynamics in the motion generation process, they are prone to the modeling errors resulting from the self-motion and have to rely on the feedback control during the execution of motion. On the contrary, the multiobody dynamics based approach often relies on the notion of the Zero Moment Point (ZMP) and very accurate models of the system dynamics [6], [7], [8], [3]. The motion in this approach is generated by defining the ZMP reference, and forcing the controller to create the CoM - often approximated by waist - trajectory that would result in the real ZMP following the reference. The motion generated with these approaches is very stable in the environments close to the ones included in the model. In most of the approaches the ZMP trajectory generation does not include information about the present system state and system dynamics, which requires prior knowledge of the future ZMP in order to generate the waist trajectory that would result in small ZMP error and not diverge in the long term, thus is difficult to use for the real-time pattern regeneration. Nevertheless, it was successfully applied in some solutions [9].

The process of pattern generation is often divided into multiple steps. Most of the approaches start from determination of the steps position based on the desired locomotion direction and ground shape. The feet position is determined geometrically [10], [11], numerically [12] or with other methods usually not taking into account the system dynamics. Then in the further stage the dynamic controller is responsible for finding the trajectory of CoM or waist which would result in a stable motion, given the predefined steps.

The method we propose in the paper is merging the simplified model and multibody dynamics based approaches. It provides the tool that takes into account the natural dynamics of the system from the very beginning when planning the foot placement and timing, and accounts for the self-motion of the multi-degree-of-freedom system. The short generation time and inclusion of the natural dynamics allows for the very rapid and dynamic motion regeneration. The short calculation time allows for the on-line implementation, while the consideration of the present dynamic state in the pattern generation process makes it a good candidate for the real-time pattern generation.

We start our paper from a short overview of the whole system. In chapter 3.we explain the simple model used in the first stage of the pattern generation and follow it with explanation of the dynamics filter used to compensate for the robot's self-motion. In chapter 5we describe how the method can be used to regenerate the gait pattern and finally in chapter 6we present patterns generated with the proposed method and results from experiments performed on our robotic platform WABIAN-2R (Fig. 1-a).

2. Two-stage Pattern Generation

In our approach we propose a two-stage pattern generation method that considers the system dynamics throughout the process (See (Fig. 1-b)). The first stage comprises a relatively simple model of a humanoid robot that allows fast and easy locomotion planning with use of conventional control theories or the motion equations. This stage of the pattern generation results in the preliminary motion references that would result in stable walking if the robot was just a single mass. In case of our implementation these are the trajectories of feet, hands and CoM. The objective of the second stage is to simulate the gait pattern resulting from the first stage on an accurate dynamic model of the system. This stage checks how the self-motion (motion of legs, arms and trunk) affect the stability of the gait and if necessary modifies the reference to improve the stability.

The main advantage of the two-stage approach, is that we can use simple mechanisms to generate the gait which follows a given reference and is consistent with the natural dynamics, yet after passing the second stage we are sure that the resultant gait will be stable when executed on the multi-degrees-of-freedom robot. The overall scheme is presented on Fig. 1-b.

In the following sections we introduce the consecutive stages of the process. The model and methodologies used inside serve as an example and can be modified and developed to account for different factors of motion dynamics or shape and properties of the environment.

3. Simple Humanoid Model

In the first stage of our pattern generator we are interested in the planning of the CoM trajectory that would follow the desired trajectory, yet stay consistent with the natural dynamics. There are three means of affecting the motion of the robot's CoM: 1) modifying the step position and timing, 2) exerting moment around the contact point with the ground and 3) controlling robot's self-motion that utilizes body's angular momentum. Since the scope of this paper is to prove the concept of the two stage pattern generation scheme we focus on the simple model that uses step position and timing to control the CoM trajectory. Thus, in this stage we use the Linear Inverted Pendulum (LIP) model presented on Fig. 2-a (for details and derivation see [13]) and derive equations that are necessary for our control strategy. The motion equations are linear and allow simple manipulation yet the output follows the tendency of the system's natural dynamics. The assumptions used in the derivation of the motion equations are as follows:

- The whole system is represented by a single mass inverted pendulum with the mass placed at the height of the robot's CoM in the free standing configuration.
- The CoM motion is constrained to the horizontal plane (the height of CoM is constant).
- There is no torque acting between the system and the ground contact point.
- The friction is big enough to prevent slipping of the foot.

Thanks to the assumptions the motion in sagittal and coronal plane can be considered separately (decoupled) and expressed with the following linear equation:

$$\ddot{y} = \frac{g}{z} y, \tag{1}$$

where, z is the height of the CoM, g is the gravitational acceleration and y is the position of CoM in the foot coordinate frame. The foot coordinate frame is attached to the sole of the foot at the projection point of the ankle joint, with initial orientation aligned with the world coordinate frame. In this paper we use y to denote position of CoM in coronal plane and x to express the position in the sagittal plane. Both are expressed with respect to the current ground contact point of the inverted pendulum (foot coordinate frame). It is worth noting that the motion does not depend on the mass of the robot, but only on the height of the CoM.

In the following subsections we describe the motion equations necessary to plan the foot placement and its timing in order to achieve the desired CoM motion. We

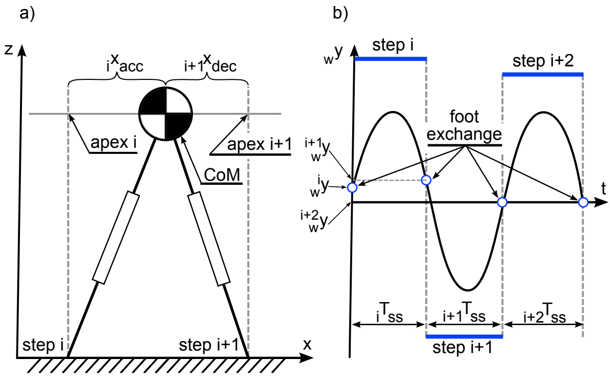

Fig. 2 a) LIP model in sagittal plane with definition of *apexes*, x_{acc} and x_{dec} in consecutive steps. b) CoM trajectory in coronal plane with respect to time. The blue lines denote step position, while the blue circles denote the moment of the foot exchange in LIP model. ${}_{w}y$ denotes coronal plane position of CoM expressed in world coordinate frame.

show example of the use of equations in deriving the motion in sagittal and coronal plane and finally describe how we generate the preliminary motion references.

3.1 Fundamental Equations

There are two sets of fundamental equations that can be derived from LIP model and used to calculate the foot placement necessary to obtain the desired CoM motion. The first set comprises equations describing position and velocity of the CoM in the support foot coordinate frame derived by integrating (1), as shown in [14].

$$y(t) = y_0 cosh(\frac{t}{k}) + \dot{y}_0 k sinh(\frac{t}{k}), \tag{2}$$

$$\dot{y}(t) = y_0 \frac{1}{k} sinh(\frac{t}{k}) + \dot{y}_0 cosh(\frac{t}{k}), \tag{3}$$

where, y_0 and $\dot{y}_0$ are the initial position and velocity of CoM with respect to the support foot, t is the time that elapsed from the initial configuration, $k = \sqrt{z/g}$.

The second set comprises one equation that is the so-called *Orbital Energy* [14]. It expresses the kinetic energy of the unit mass inverted pendulum. It is calculated by integrating (1) over the traveled distance and has the following form:

$$E = \frac{1}{2}\dot{y}_1^2 = \frac{g}{z}(y_1^2 - y_0^2) + \frac{1}{2}\dot{y}_0^2, \tag{4}$$

where, y_1 and $\dot{y}_1$ are the position and velocity of the CoM at the end of the motion.

3.2 Generation of Motion in Sagittal and Coronal Plane

In this subsection we describe how using (2)-(4) we generated the reference foot steps to achieve the desired CoM motion.

First we start from defining the equivalent of *apex* in case of the LIP model (See Fig. 2). We define apex to be the point along the CoM trajectory in the considered plane of motion where CoM reaches the lowest speed. In case when the CoM passes over the support foot (sagittal plane) the *apex* will be placed above the foot. In case when the CoM does not reach the support foot and starts falling back (coronal plane), *apex* will be the point where CoM velocity reaches 0.

A. Sagittal plane

Taking a closer look at the trajectory of CoM during the single support phase, we can notice that the CoM decelerates before reaching the *apex* and accelerates after passing the *apex* point. By controlling the distance of acceleration during one step and deceleration during the consecutive step we can control the change of velocity of CoM in the consecutive *apexes*. We can derive the equation governing the relation between the two distances and the step length from the *Orbital Energy* equation (4). For derivation purposes we define the desired step length as x_{sl} and assume that it is given. We also define the ratio between the CoM acceleration distance x_{acc} and step length x_{sl} as:

$$r = \frac{x_{acc}}{x_{sl}}. \tag{5}$$

The following equation expresses CoM velocity in *apex* $i + 1$:

$${}^{i+1}\dot{x}_{apex} = {}^{i}\dot{x}_{apex} + \frac{z}{2g}({}^{i}x_{acc}^2 - {}^{i+1}x_{dec}^2), \tag{6}$$

where, the left superscript denotes the step number, this notation is used from here on. By noting that ${}^{i}x_{sl} = {}^{i}x_{acc} + {}^{i+1}x_{dec}$, substituting (5) into (6) and simplifying we obtain the following equation:

$$r = \frac{g}{z}\frac{{}^{i+1}\dot{x}_{apex}^2 - {}^{i}\dot{x}_{apex}^2}{2{}^{i}x_{sl}^2} + \frac{1}{2} \tag{7}$$

Given the step length, (5) and (7) we can calculate the acceleration and deceleration distances. The acceleration distance determines the time remaining to the foot exchange from the previous *apex*, while the deceleration distance determines the distance between the CoM and the foot position during the foot exchange. Repeating the calculation for each consecutive step, given its *apex* reference velocity and desired step length, we are able to calculate the motion of CoM, foot steps and the single support time in the sagittal plane.

B. Coronal plane

The motion in coronal plane, unlike sagittal, comprises of CoM swings that should never cross the support foot position (Fig. 2-b), unless the robot is subjected to very high disturbances. Planing motion in coronal plane is reduced to determining the step position so that the CoM does not cross the foot and the single support time has the desired duration.

By starting the motion generation from the sagittal plane, we predefine the single support time for each step of the gait. Thus, to plan the coronal plane CoM trajectory of each step we need to find such foot placement that will result in the same single support time as in sagittal plane. Apart from the step time, we also want to be able to control the step width as that affects the lateral sway of CoM and is constrained by the movable range of the robot. One last parameter that we want to control is the position of CoM in the world coordinate frame during the foot exchange, since this will let us to control the locomotion trajectory of the robot Fig. 2-b.

Assuming that the *apex* velocity reference in sagittal plane is constant over a number of steps, the single support time (T_{ss}) during these steps will also be constant and the system will be in a limit cycle. Now, given arbitrary initial conditions we would like to bring our LIP model into the limit cycle in the coronal plane, in which the initial CoM position y_0 at the moment of foot exchange is equal to the half of the step width and the CoM position in the world coordinate frame is equal to ${}_{w}y_0$ (here, the left subscript denotes that the value is expressed in the world coordinate frame). The initial velocity should be determined so that after the time T_{ss} elapses, the CoM will be in exactly same position y_0. To calculate the initial velocity we can use formula (3) knowing that for the above to be true the velocity after $T_{ss}/2$ should be zero.

$$
{}^{i}\dot{y}_0 = -{}^{i}y_0 k \frac{sinh({}^{i}T_{ss}/2k)}{cosh({}^{i}T_{ss}/2k)}, \tag{8}
$$

where, ${}^{i}T_{ss}$ denotes single support time in step i. In order to achieve this velocity and to assure that the CoM position in the world coordinate frame during foot exchange is at the desired place, when starting from arbitrary initial conditions, we need to control the step position of two preceding steps. By combining (2) and (3) for two preceding steps and adding condition for CoM being at the desired global position during foot exchange we arrive at the following set of equations.

$$
\begin{cases}
{}^{i+1}y({}^{i+1}T_{ss}) = {}^{i+1}y(0)\,{}^{i+1}A + {}^{i+1}\dot{y}(0)\,{}^{i+1}B \\
{}^{i+1}\dot{y}({}^{i+1}T_{ss}) = {}^{i+1}y(0)\,{}^{i+1}C + {}^{i+1}\dot{y}(0)\,{}^{i+1}A \\
{}^{i}\dot{y}({}^{i}T_{ss}) = {}^{i}y(0)\,{}^{i}C + {}^{i}\dot{y}(0)\,{}^{i}A = {}^{i+1}\dot{y}(0) \\
{}^{i+2}_{w}y(0) - {}^{i}_{w}y(0) = \\
-{}^{i}y(0) + {}^{i}y({}^{i}T_{ss}) - {}^{i+1}y(0) + {}^{i+1}y({}^{i+1}T_{ss})
\end{cases} \tag{9}
$$

where, ${}^{i}A = cosh({}^{i}T_{ss})$, ${}^{i}B = k \cdot sinh({}^{i}T_{ss})$ are ${}^{i}C = sinh({}^{i}T_{ss})/k$ are elements of equation (2) and (3). The simplification yields a set of linear equations, whose solution provides the desired step position in two consecutive steps preceding the step of interest (here step number $i + 2$).

$$
\begin{bmatrix} {}^{i}y(0) \\ {}^{i+1}y(0) \end{bmatrix} = \begin{bmatrix} {}^{i}C^{i+1}B + {}^{i}A - 1 & {}^{i+1}A - 1 \\ {}^{i}C^{i+1}A & {}^{i+1}C \end{bmatrix}^{-1} \times
\begin{bmatrix} {}^{i+2}_{w}y(0) - {}^{i}_{w}y(0) - {}^{i}\dot{y}(0)({}^{i}B + {}^{i}A^{i+1}B) \\ {}^{i+1}\dot{y}({}^{i+1}T_{ss}) - {}^{i}\dot{y}(0)^{i}A^{i+1}A \end{bmatrix} \tag{10}
$$

We use the above equations to calculate the step position in every single step calculation. The method proved to be stable in a variety of tested scenarios. Since the above equations do not include the system limitations, like feasible workspace, the result has to be verified against them and if necessary limited. Since the step position is recalculated at each foot exchange based on the current state, even after limiting the step position the robot will reach the desired global position in the following steps.

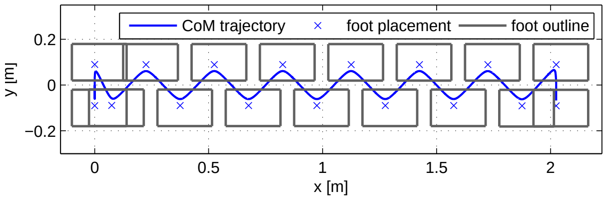

Fig. 3 Top view of the CoM trajectory and the foot placement resulting from the LIP model. The limit cycle step size: 0.15m; the limit cycle step width: 0.19m.

3.3 Starting the Motion

Since the methods described in 3.2 assume that the system is already in motion, we need to put it into motion by specifying initial CoM trajectory.

A. Sagittal plane

We initiate the motion by applying the virtual torque to the ankle joint until the mass reaches the desired x_{acc}. The trajectory of the CoM is calculated with modified version of (1) to include the effect of the torque.

$$
\ddot{x} = \frac{g}{z}x + \frac{\tau_y}{mz} \tag{11}
$$

B. Coronal plane

We start from generating initial swing in the direction of the first swing foot. The swing is calculated in a way that will make the CoM reach the desired velocity when crossing the sagittal plane (middle) on the way back towards the first support foot. The desired velocity is calculated so that the time needed for CoM to return back to the sagittal plane is equall to ${}^{1}T_{ss}$.

3.4 Generation of the Preliminary Motion Reference Trajectories

The example of the trajectory generated with the method described above is presented on Fig. 3. In order to prepare for the second stage of the pattern generation, we need to calculate the motion references that predefine the motion of the whole robot in space. To define the configuration of the robot we need to specify the feet, hands and waist trajectories. The feet trajectories are generated with use of the single fifth-order polynomial in x and y direction, and two separate polynomials in z direction for the rising and lowering of the foot. Since the LIP model does not include the double support phase, we artificially accelerate the landing time and prolong the lift off time to generate the double support phase. This time is subtracted from the original single support time. For simplification we set the hands trajectories by adding a constant vector to the waist trajectory, in order to fix them in place. The initial waist trajectory is calculated in the first iteration of the Preview Controller.

4. Dynamics Filtering

The second stage of the proposed pattern generator comprises the dynamics filter. The dynamics filtering was first used in humanoid robotics by Kajita in [8] to refine the motion pattern resulting from the first iteration of his Preview Controller. We decided to call the second stage of our scheme dynamics filtering, because we verify the generated motion reference with detailed dynamic model of the robot. If the discrepancy between the two models is too big, the motion reference is modified to improve the stability of the locomotion of the multi-degree-of-freedom robot. The dynamics filter can comprise of any gait controller that accepts as input the motion reference and uses the Multibody System (MBS) dynamics to refine the reference in order to make the gait stable. One of the common motion references is the Zero Moment Point (ZMP) and there are two well known methods utilizing the MBS inside the pattern generation process: Preview Control developed by Kajita et al. [8] and FFT based pattern generator developed by Takanishi et al. [3].

In this implementation we used the Preview Controller, however any other method utilizing MBS in the control loop could be used in this stage. In the next subsections we describe how do we obtain the ZMP reference and explain the basic principles of the Preview Controller.

4.1 Calculating ZMP Reference

To use the ZMP based method we first need to obtain the ZMP reference. We obtain the ZMP information from the results of the first pattern generation stage. Given the foot position and CoM position and acceleration in time we can calculate the ZMP with the following formula.

$$ZMP_x^{ref} = {}_w x - \frac{z}{g}\ddot{x} \tag{12}$$

The result will exactly follow the position of the support foot in time given the LIP model is used in the first stage. We decided to use the above equation in case we extend the model used in the first stage. In that case the dynamics filter does not need any modifications and the ZMP reference calculated with use of (12) includes all the factors affecting the CoM motion.

After calculating the ZMP, we apply the low pass filter to smooth the reference and include the smooth transition between the feet during the double support phase.

4.2 Preview Control

We do not intend to present the derivation and details of the Preview Control, instead we try to explain its basic working principles. Readers interested in details of the Preview Controller should refer to [8].

The task of the preview controller is, given the ZMP reference, to generate the CoM trajectory of the robot that will result in the ZMP resulting from MBS simulation matching the reference ZMP with the specified error. The input comprises ZMP trajectory and the reference trajectories which pre-define the motion of the robot in 3D space. In our case, these are the feet and hands trajectories, the waist trajectory is determined and later modified by the preview controller. The flow-chart of the preview control is presented on Fig. 4. In the first iteration of the Preview Control the initial waist trajectory in sagittal and coronal plane separately is calculated with the state space equation of which the controlled variable is expressed with the following equation:

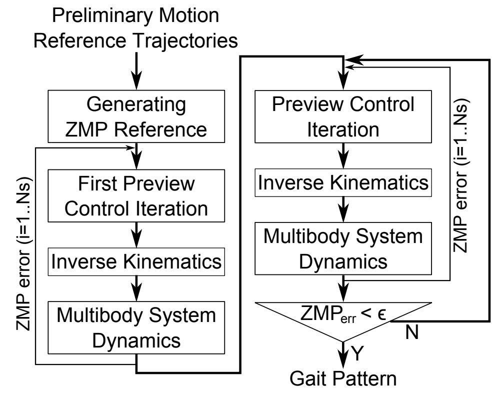

Fig. 4 The flow of the dynamics filter with preview control. Ns denotes the total number of reference samples in the single pattern.

$$u(i) = -G_i \sum_{j=0}^{i} e(j) - G_x \boldsymbol{x}(i) - \sum_{j=1}^{N_L} G_p(j) ZMP_x^{ref}(i+j), \tag{13}$$

where, G_i is an integral gain trying to compensate for the steady state ZMP error, G_x is a matrix of proportional gains serving as a state feedback and G_p is a vector of preview gains which modify the control variable depending on the future ZMP reference, $e(i)$ is an error between the reference ZMP and the ZMP calculated with MBS model in the i^{th} sample of the trajectory, $\boldsymbol{x}$ is the state vector containing position, velocity and acceleration of waist in sagittal plane, N_L is the number of preview samples and ZMP_x^{ref} is the ZMP reference trajectory. This iteration provides the initial waist trajectory and the ZMP error between the original reference and the MBS model. In the following iterations of the Preview Controler the waist trajectory is modified based on the ZMP error from the previous iterations, according to the following equation

$$u(i) = -G_i \sum_{j=0}^{i} e(j) - G_{\Delta} \boldsymbol{x}(i) - \sum_{j=1}^{N_L} G_p(j) e^{prev}(i+j), \tag{14}$$

where, $\Delta \boldsymbol{x}$ is the state space correction vector and e^{prev} is the ZMP error in the previous iteration. This iteration is repeated either until the ZMP error falls bellow a given threshold or given number of iterations. In our case, just two iterations were sufficient to reduce ZMP error below 0.003m for the regular forward walk.

The waist trajectory generated in the last iteration, together with remaining motion references become the final gait pattern of the generator.

5. Pattern Regeneration

The method we described in the two previous sections can be applied to regenerate the gait pattern from arbitrary point of the swing phase. The point where the new trajectory is going to be connected serves as a source of initial conditions of both the LIP model in the first stage of the pattern generation and the Preview Controler in the second. In this section we describe the changes and considerations that need to be taken into account when regenerating the pattern.

5.1 Simplified Model

Since the regeneration of the pattern starts when the robot is in motion there is no need to generate the starting motion and we can proceed to planning motion as described in 3.2 Inside the first stage we calculate each step individually, so the only difference between the algorithm described in 3.2 is the calculation of the first step. The initial conditions for the first step, namely the position and velocity of CoM with respect to the support foot, are taken from the reference calculated in the first stage of the original pattern generation.

A. Sagittal plane

For generation of the sagittal plane motion we can use (2), (3) and (7). Since the starting point is not an *apex* we need to modify (7) used to calculate ratio between the acceleration and deceleration to include the starting position and velocity of the CoM.

$$r = \frac{g}{z}\frac{{}^{i+1}\dot{x}^2_{apex} - {}^{i}\dot{x}^2_0}{2^i x^2_{sl}} + \frac{x^2_0}{2^i x^2_{sl}} + \frac{1}{2}, \qquad (15)$$

where, x_0 is the position and $\dot{x}_0$ is the velocity of CoM at the pattern connection point. We have to ensure that the result fulfills $r > abs(\frac{x_0}{x_{sl}})$, since the acceleration distance cannot be smaller than the initial CoM position.

B. Coronal plane

The generation of motion in the coronal plane can remain the same, since the initial conditions used in the calculation of the step position are taken from the expected foot exchange point. As we regenerate the motion always starting from the double support point, knowing the current conditions and the remaining duration of the single support phase we can calculate conditions during the foot exchange with (2) and (3). The example of the regenerated pattern is presented in 6.

C. Preliminary motion reference trajectories

Feet trajectory is regenerated by specifying the first step initial condition of the polynomials equal to the foot position, velocity and acceleration at the pattern connection point of the original trajectory and the final conditions according to the LIP model output. In case the foot needs to be raised again we modify only the initial acceleration in the z direction.

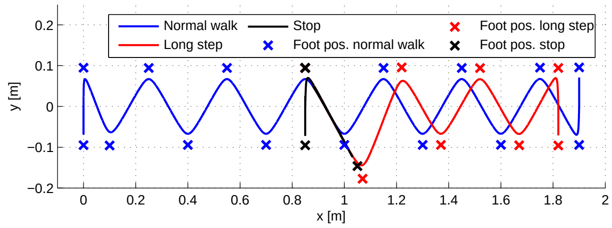

Fig. 5 CoM trajectories and foot steps resulting from the first stage of the pattern generator for the three scenarios. (Note: plot aspect ratio is not 1:1)

5.2 Dynamics Filtering

The only necessary modification to the preview controller is the initial value of the state vector. In this case we use the value at the connection point, obtained from the Preview Controler when generating the original pattern.

6. Output Patterns and Experiments

In this section we present an example of the normal walking pattern and two examples of its very fast regeneration simulating the sudden intrusion of an object into the robot's path or sudden detection of a hole in the ground. The experiments were performed in order to verify how the robot behaves after connecting newly generated pattern which introduces sudden change of movement. The pattern is regenerated off-line and connected to the selected pattern connection point. To simulate the on-line calculation, during the regeneration process we assume that we can modify the reference only after the chosen pattern connection point. The total computation time of the 10s pattern given the 10ms sampling time takes under 30ms under Linux Ubuntu operating system on single core of 2.10GHz i7 processor.

6.1 Regular Pattern

The regular pattern is created by setting the reference data for the first stage pattern generator to the following values:

- apex velocity: 0.082 m/s
- desired step width: 0.19 m
- number of steps: 14

The plot of the resultant CoM trajectory in world coordinate frame is presented on Fig. 5.

6.2 Sudden Stop

In this scenario we simulate the sudden intrusion of a big object into the path of the robot. The only way to avoid the collision is to stop the motion and back out. In this case we regenerate the pattern from the point 140ms before the ground contact of the right foot. The connection point was chosen as close to the ground contact as possible, but far enough to avoid reaching the joint velocity limits during sudden movement of the foot. The information given to the generator is the negative refer-

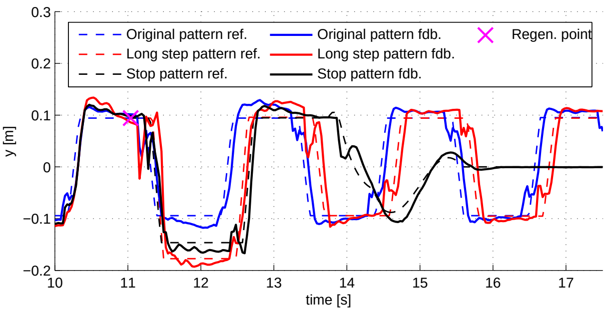

Fig. 6 ZMP reference and feed-back data in coronal plane registered during experiments.

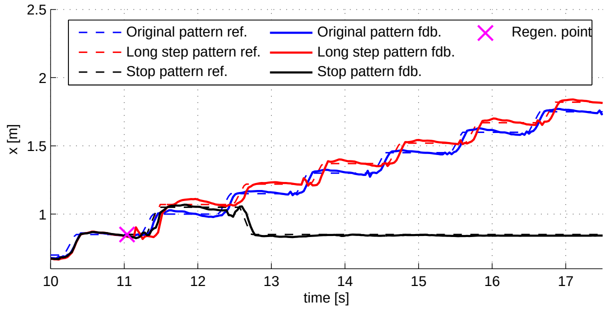

Fig. 7 ZMP reference and feed-back data in sagittal plane registered during experiments.

ence *apex* velocity and the information that in the first step CoM should reach *apex* 2cm before the foot position. Also, since the robot is already very close to the foot exchange point it is not possible to slow down motion given the current step size, thus we increase the step size by 5cm (the distance was chosen empirically, but can be calculated if necessary). Fig. 5 shows the regenerated CoM trajectory. We can notice that the first stage model automatically modified the position of the foot in the coronal plane to account for the change in the single support time. The Fig. 6 and 7 show the ZMP reference and the feed-back data, of several steps before and after the regeneration point, registered during experiments. One can notice that the ZMP error is not significantly bigger during the sudden foot motion than in other part of the pattern. Also from the reference ZMP trajectory we can notice how the LIP model is modifying the stance time to balance the velocity at the *apex*.

6.3 Sudden Change of the Step Size

In this scenario we simulate the sudden detection of the hole or other flat obstacle that forbids the robot from placing the foot in the originally planed position. The only way to avoid tipping of the robot is to quickly change the placement of the foot. In this scenario we regenerate the pattern from the same point (140ms before ground contact). The only major change of the reference sent to the first stage generator, compared to the normal walk, is the new length of the first step. One can see on Fig. 5 that the first stage pattern generator automatically tuned the position of the foot in coronal plane to account for the change in the single support time. The change of the foot placement results in the very sudden motion of the foot (0.11m in 230ms). Fig. 8 shows the ZMP reference and the errors of the regenerated pattern in the individual Preview Control iterations. One can notice that the sudden motion of the foot results in the big ZMP error which cannot be accounted for in the LIP model, but is corrected in the consecutive Preview Control iterations of the second stage of the pattern generation.

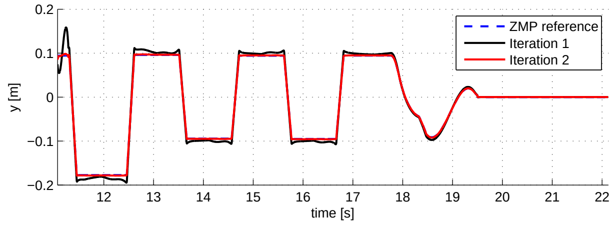

Fig. 8 Regenerate ZMP reference trajectory and ZMP trajectories calculated with Multibody System during pattern regeneration process.

7. Summary

This paper is a proof of concept of the proposed two-stage pattern generation scheme. The first stage is based on the inverted pendulum model that allows for easy and autonomous generation of the step placement and the CoM trajectory consistent with the natural dynamics of the robot. The model used to present the method is constrained to the forward locomotion on the flat ground, but it can be easily substituted with more advanced models including the ground contact and angular momentum effects or models enabling locomotion in 3D or in environments with different mechanical properties. The second stage is based on the Preview Control used as a dynamics filter that fine tunes the reference trajectory to improve the overall gait stability for execution on multi-degree-of-freedom robots. This controller also can be substituted with any controller using the Multibody System dynamics in the loop. We show that using the proposed scheme we are able to easily generate and regenerate gait pattern which is consistent with natural dynamics and stable even in case of the very abrupt motion (for the better image of the motion we encourage the reader to view the video of the experiment[15]). The method was tested by generating the pattern off-line and executing it on the robot. The information and constraints used for regeneration, however, were the same as if the code was implemented on-line. Because the execution time is very short ($\sim$30ms) and the regeneration of the pattern is based on the dynamic state of the system at the regeneration point, we believe that the method has a very high potential of being implemented as the real-time controller and this will be our next objective.

References

[1] S. Kajita and K. Tani, “Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode,” in *Proc.*

ICRA Robotics and Automation IEEE Int. Conf, 1991,

[2] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *Humanoid Robots, IEEE-RAS International Conference on*. IEEE, 2006,

[3] J. Yamaguchi, E. Soga, S. Inoue, and A. Takanishi, "Development of a bipedal humanoid robot-control method of whole body cooperative dynamic biped walking," in *Proc. ICRA Robotics and Automation IEEE Int. Conf*, vol. 1, 1999,

[4] V. Hugel and N. Jouandeau, "Walking patterns for real time path planning simulation of humanoids," in *RO-MAN, IEEE*, 2012,

[5] J. Englsberger, C. Ott, M. Roa, A. Albu-Schaffer, and G. Hirzinger, "Bipedal walking control based on capture point dynamics," in *Proc. IEEE/RSJ Int Intelligent Robots and Systems (IROS) Conf*, 2011,

[6] P.-B. Wieber, "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," in *Humanoid Robots, IEEE-RAS International Conference on*. IEEE, 2006,

[7] K. Harada, S. Kajita, K. Kaneko, and H. Hirukawa, "An analytical method on real-time gait planning for a humanoid robot," in *Humanoid Robots, IEEE/RAS International Conference on*, vol. 2. IEEE, 2004,

[8] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, and K. H. K. Yokoi, "Biped walking pattern generation by using preview control of zero-moment point," in *Proc. ICRA Robotics and Automation IEEE Int. Conf*, 2003,

[9] K. Nishiwaki and S. Kagami, "Strategies for adjusting the zmp reference trajectory for maintaining balance in humanoid walking," in *Robotics and Automation (ICRA), IEEE International Conference on*, 2010,

[10] K. Nishiwaki, S. Kagami, J. J. Kuffner, M. Inaba, and H. Inoue, "Online humanoid walking control system and a moving goal tracking experiment," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 1. IEEE, 2003,

[11] H. Kobayashi and T. Sugihara, "Self-consistent automatic navigation of com and feet for realtime humanoid robot steering," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE, 2009,

[12] J. Chestnutt, K. Nishiwaki, J. Kuffner, and S. Kagami, "An adaptive action model for legged navigation planning," in *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*. IEEE, 2007,

[13] 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," in *Proc. IEEE/RSJ Int Intelligent Robots and Systems Conf*, vol. 1, 2001,

[14] S. Kajita, O. Matsumoto, and M. Saigo, "Real-time 3D walking pattern generation for a biped robot with telescopic legs," in *Proc. ICRA Robotics and Automation IEEE Int. Conf*, vol. 3, 2001,

[15] "Video accompanying the paper, available online: http://kormushev.com/goto/URAI-2013."