Online Regeneration of Bipedal Walking Gait Optimizing Footstep Placement and Timing

Przemyslaw Kryczka[1], Petar Kormushev[1,2], Nikos G. Tsagarakis[1] and Darwin G. Caldwell[1]

***Abstract*— We propose a new algorithm capable of online regeneration of walking gait patterns. The algorithm uses a nonlinear optimization technique to find step parameters that will bring the robot from the present state to a desired state. It modifies online not only the footstep positions, but also the step timing in order to maintain dynamic stability during walking. Inclusion of step time modification extends the robustness against rarely addressed disturbances, such as pushes towards the stance foot. The controller is able to recover dynamic stability regardless of the source of the disturbance (e.g. model inaccuracy, reference tracking error or external disturbance).**

We describe the robot state estimation and center-of-mass feedback controller necessary to realize stable locomotion on our humanoid platform COMAN. We also present a set of experiments performed on the platform that show the performance of the feedback controller and of the gait pattern regenerator. We show how the robot is able to cope with series of pushes, by adjusting step times and positions.

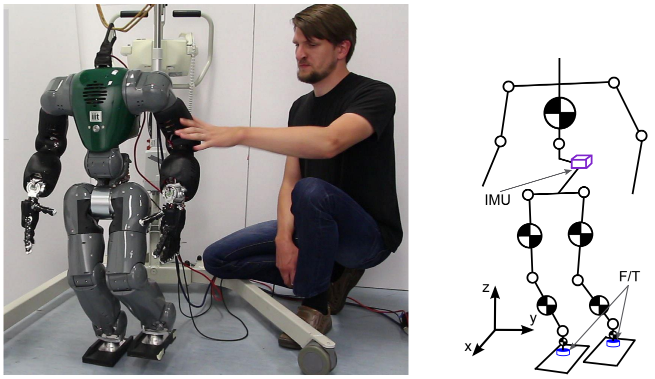

Fig. 1. Left image shows extract from experiment in which our humanoid research platform COMAN is being pushed towards the stance foot. The drawing on the right shows the mass distribution of our MBS model and indication of the IMU used in our state estimator.

I. INTRODUCTION

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 inaccuracies in the models or external disturbances.

The research on humanoid robots significantly intensified in recent years producing numerous methods for disturbance rejection and quick gait pattern regeneration. Diedam et al. developed a Linear Model Predictive Controller which includes the foot step planning in the optimization task, thus enabling generation of reactive steps in case of disturbance [1], [2]. Nishiwaki et al. proposed a series of methods of quick gait pattern regeneration [3] and motion replanning strategies for walking on rough terrain [4]. Morisawa et al. developed a method of simultaneous planning of center of mass (COM) and zero moment point (ZMP) with use of numerical optimization method. The controller enabled disturbance rejection by changing the step position and ZMP shaping [5]. Urata et al. derived an explicit solution of modified version of preview controller and used it in real-time optimization of gait pattern for disturbance rejection [6].

Most of the existing disturbance rejection methods assume a constant step time or allow only its minimal change when replanning the motion. The major corrective actions are realized by modifying the step placement. This however limits the kind of disturbances the robots are able to handle. Especially when the disturbance acts in the direction of the present stance foot, which can be only realized by changing the step time or by crossing the legs. The latter action is however mechanically risky or impossible due to self-collision between the legs for most of the existing humanoid platforms. Humans when subjected to external disturbance or unexpected change in the ground conditions manipulate not only the step position, but also a step time.

In this paper we describe a novel method which enables optimized simultaneous selection of all the three fundamental parameters affecting the gait, namely step position in sagittal and coronal plane as well as the step time, thus extending the kind of disturbances the robot is resistant to. Given the estimated COM position and velocity we use a nonlinear optimization method to find optimized step positions and step timing for the present and proceeding two steps. For calculation of the remaining steps we use previously developed method [7]. The gait pattern regeneration time which includes solving optimization problem and generation of gait pattern with Multi Body System (MBS) in the loop takes less than 40 ms. The experiments we performed on our research platform COMAN [8](Fig. 1) prove that the method is able to cope with disturbances and that the algorithm controls both step position and step time. In the paper, together with

[1] Department of Advanced Robotics, Istituto Italiano di Tecnologia (IIT), Via Morego 30, 16163 Genoa, Italy.
Contact e-mail: przemyslaw.kryczka@iit.it

[2] Dyson School of Design Engineering, Imperial College London, South Kensington campus, London SW7 2AZ, United Kingdom

* The research leading to these results received funding from the FP7 European project WALK-MAN (ICT-2013-10) and the ECs Horzion 2020 robotics program ICT-23-2014 under grant agreement 644727 - CogIMon.

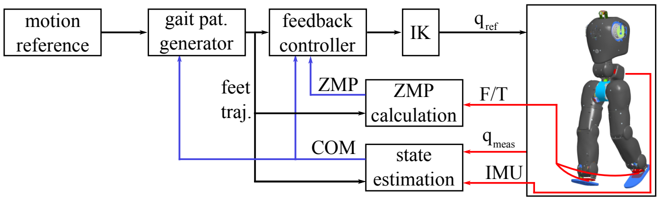

Fig. 2. Overview of the locomotion control system. The outer gait pattern generator is able to regenerate the pattern every 40 ms, while the inner feedback control works with the same frequency as joint controller which is 1 kHz.

elaboration of the gait pattern generator, we describe the state estimation method and the COM feedback controller, which is necessary for stable locomotion on this compliant platform.

The paper comprises of four sections. In the next section we explain details of the gait pattern generation method. In section III we explain the way we estimate the state of the robot and the feedback controller we use for stabilizing locomotion of the robot. Finally in the last section we describe experiments performed on our robotic platform COMAN.

II. PATTERN GENERATION

In this section we will describe the details of our gait pattern generation method. We will start from a short overview of the locomotion control system and show the place of the gait pattern generator inside the system and then proceed to discuss its details .

A. Locomotion control overview

In our locomotion controller (Fig. 2) we take an average locomotion velocity as an input. Based on this we generate the first gait pattern based on a method described in [7]. This stage results in reference of the feet, pelvis and COM trajectories. The reference is pulled every 10 ms from gait pattern generator for execution on the robot. Since COMAN is an intrinsically compliant robot, the pattern is not exactly followed and feedback control is needed to perform a stable locomotion. The details of our feedback controller are described in Sec. III. Finally the pattern modified by feedback controller is executed on the robot.

During pattern execution we keep comparing the reference and estimated position of COM to check for deviations. If the error in COM position is bigger than certain threshold (in our case 0.01 m) we regenerate the gait pattern starting from the estimated configuration. The details of the gait pattern regeneration method are described in the following section.

B. Gait pattern regeneration method

When the estimated motion of the robot diverges from the pre-calculated pattern we regenerate the locomotion trajectory to keep it stable. In order to better exploit the dynamic capabilities of the system, we exploit all gait parameters affecting the locomotion, namely position of the step in sagittal and coronal plane, as well as the step time. Since the motion in sagittal and coronal plane - given the assumption of the Linear Inverted Pendulum (LIP) model that the COM does not move in vertical direction - are coupled in time, we formulate the problem in a way that allows us to choose those parameters, taking into account their interdependence. Since the COM trajectory is a nonlinear function of time and of initial conditions, we decided to solve the problem with use of a nonlinear optimization technique. Our problem is defined as

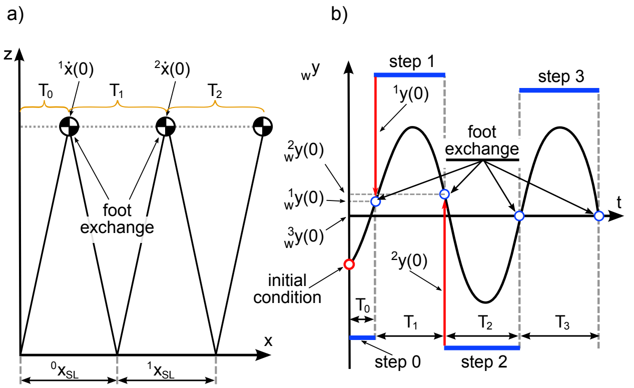

Fig. 3. Drawing a) shows the sagittal plane of the LIP model and used nomenclature. Drawing b) shows coronal plane trajectory of COM in time and related nomenclature. Note that the left subscript w denotes the world coordinate frame. In many of the equations the COM position in coronal plane is expressed in foot coordinate frame. It is also worth noting that the first time T_0 is a remaining step time, not the full step time.

$$\min_{\mathbf{v}\in\mathbb{R}^n} \quad f_{opt}(\mathbf{v}) \tag{1}$$

$$s.t. \quad \mathbf{v}^L \leq \mathbf{v} \leq \mathbf{v}^U \tag{2}$$

where $\mathbf{v}$ is a vector of optimization variables and $\mathbf{v}^L$ and $\mathbf{v}^U$ are its lower and upper bounds respectively. We denote optimization variables with $\mathbf{v}$ instead of usual $\mathbf{x}$ to avoid ambiguity between COM position in the sagittal plane. $f_{opt} : \mathbb{R}^n \longrightarrow \mathbb{R}$ is a nonlinear scalar cost function. The task of the optimization problem is to find step positions and timing necessary to bring COM motion into the steady gait (constant step size and time). To enter the steady gait, COM needs to enter the step with given velocity in both sagittal and coronal plane. To bring it from any initial position and velocity to a given position and velocity at the step onset we need to control position of two consecutive steps. To increase robustness of the method we could use more than two steps, but it would significantly increase the calculation time. The variables are spread between the directly controlled optimization variables and variables embedded in cost function. In our problem formulation there are four optimization variables defined as:

$$\mathbf{v} = [T_0 \; T_1 \; T_2 \; {}^0x_{SL}]^T \tag{3}$$

where T_0 to T_2 are the present remaining step time and two proceeding step times and ${}^0x_{SL}$ is a present step length. The bounding vectors $\mathbf{v}^L$ and $\mathbf{v}^U$ limit the range of possible step times and present step length. In our experiments the values were:

$$\mathbf{v}_L = [0.3 \; 0.6 \; 0.6 \; 0]^T; \quad \mathbf{v}_U = [1.5 \; 1.5 \; 1.5 \; 0.15]^T \tag{4}$$

Our cost function is defined as:

$$f_{opt}(\mathbf{v}) = \mathbf{w}^T\mathbf{f}(\mathbf{v}) \tag{5}$$

where $\mathbf{f}(\mathbf{v})$ is a vector of cost function components and $\mathbf{w}$ is a weight vector. The cost function is derived based on the LIP model. The vector containing cost function components is formulated as follows:

$$\mathbf{f}(\mathbf{v}) = \begin{bmatrix} ({}^{1}y(0) - {}^{1}y(0)_{ref})^2 \\ ({}^{2}y(0) - {}^{2}y(0)_{ref})^2 \\ ({}^{0}x_{SL} - {}^{0}x_{SLref})^2 \\ (T_1 - T_{ref})^2 \\ (T_2 - T_{ref})^2 \\ ({}^{2}\dot{x}(0) - {}^{2}\dot{x}(0)_{ref})^2 \end{bmatrix} \tag{6}$$

All of the components of the vector, except from time, are nonlinear functions of the optimization variables. The first two components correspond to the initial distance of the COM from support foot, right after foot exchange during two consecutive steps (Fig. 3). The distances are calculated in order to bring the COM lateral sway into limit cycle. The values are calculated with equation below. The derivation of the equation and more detailed description is presented in [7].

$$\begin{bmatrix} {}^{1}y(0) \\ {}^{2}y(0) \end{bmatrix} = \begin{bmatrix} {}^{1}C^{2}B + {}^{1}A - 1 & {}^{2}A - 1 \\ {}^{1}C^{2}A & {}^{2}C \end{bmatrix}^{-1} \times \begin{bmatrix} {}^{3}_{w}y(0) - {}^{1}_{w}y(0) - {}^{1}\dot{y}(0)({}^{1}B + {}^{1}A^{2}B) \\ {}^{3}\dot{y}(0) - {}^{1}\dot{y}(0)^{1}A^{2}A \end{bmatrix}, \tag{7}$$

The left superscripts in (7) and following equations denote step number, while the left subscript w denotes that the value is expressed in world coordinate frame. ${}^{i}A = cosh(T_i/k)$, ${}^{i}B = k \cdot sinh(T_i/k)$ and ${}^{i}C = sinh(T_i/k)/k$ are components of LIP model time domain equations in which $k = \sqrt{z_{COM}/g}$. ${}^{3}_{w}y(0)$ is a desired global position of COM after two steps expressed in the world coordinate frame (left subscript w), T_i is the i-th step time, ${}^{3}\dot{y}(0)$ is a desired COM velocity after two steps, ${}^{1}_{w}y(0)$ is an initial COM position expressed in world coordinate frame and ${}^{1}\dot{y}(0)$ is an initial COM velocity. The reference values for the steps are taken from the previously calculated pattern in order to minimize the foot displacement. Furthermore, in the experimental section the reference value for ${}^{3}_{w}y(0)$ is set to be equal to ${}^{1}_{w}y(0)$. Equation (7) takes into account the global position and velocity during the first foot exchange which are calculated with:

$${}^{1}_{w}y(0) = {}^{0}_{w}y_{step} + {}^{0}y(0)^{0}A + {}^{0}\dot{y}(0)^{0}B, \tag{8}$$

$${}^{1}\dot{y}(0) = {}^{0}y(0)^{0}C + {}^{0}\dot{y}(0)^{0}A, \tag{9}$$

where ${}^{0}_{w}y_{step}$ is a present position of the stance foot in the global coordinate frame.

The third component of (6) corresponds to the present step length, which together with step time affects the forward velocity of locomotion. The reference value of the step size is also taken from previous pattern in order to minimize the foot displacement in the new pattern.

The fourth and fifth components of (6) correspond to the two step times following the present step. These components try to bring the step time to desired value (in our experiments set to $T_{ref} = 1s$).

The last component of (6) corresponds to the initial velocity in the second step and aims at bringing the robot to the desired forward locomotion velocity starting from the second step. It is a function of step time and the first step length and is expressed with:

$${}^{0}x(T_0) = {}^{0}x(0)^{0}A + {}^{0}\dot{x}(0)^{0}B \tag{10}$$

$${}^{1}\dot{x}(0) = {}^{0}x(0)^{0}C + {}^{0}\dot{x}(0)^{0}A \tag{11}$$

$${}^{1}x(0) = {}^{0}x(T_0) - {}^{0}x_{SL} \tag{12}$$

$${}^{2}\dot{x}(0) = {}^{1}x(0)^{1}C + {}^{1}\dot{x}(0)^{1}A \tag{13}$$

It is worth noting that while the present step time T_0 is one of the optimization variables, we do not include it directly in cost function, it is however indirectly present in its subcomponents. This allows for free exploitation of the present step time in order to achieve the optimization goal.

The weight vector $\mathbf{w}$ contains the scaling factors which intend to compensate for differences in units of different components of the cost function vector. It also contains the weights which differentiate importance of each of the cost function components. Set of weights used in our experiments is following:

$$\mathbf{w} = [950\ 5\ 450\ 0.1\ 0.1\ 4.4]^T \tag{14}$$

The first and third gains are high in order to minimize displacement of foot in the present step.

This is the formulation of our nonlinear optimization problem. To solve it we use an interior-point nonlinear optimization library IPOPT available under EPL license [9]. Even though there are solvers with better performance, we chose IPOPT as it was an easily available and non-commercial nonlinear optimization solver. The calculation of the optimization problem takes on average 10 ms on a single core of an Intel Core i7 @ 2.1 GHz and depends on initial conditions.

The solution of described optimization problem provides three consecutive step times, two consecutive footsteps in coronal plane and one footstep in sagittal plane. For safety reasons every time the gait pattern is regenerated we compute more than 3 steps with gait ending in standing position. The footsteps in sagittal plane are calculated in a way to keep the terminal velocity calculated during optimization, while the steps in coronal plane are calculated according to (7). For more details on regular steps calculation refer to [7].

The initial pattern calculated this way provides motion reference for single mass robot, which is far from the humanoid platform. Thus to calculate the final gait pattern we use the COM motion resulting from previous stage to calculate the ZMP reference trajectory and use preview controller [10] with MBS (Fig. 1) in the loop to calculate the motion reference for our robot. When regenerating the pattern in this stage we also start from the initial condition of COM estimated by our state estimator. Since that creates the reference pattern that is discontinuous we interpolate between previous pattern and the new pattern over 150 ms window. More detailed description of method we use to refine the gait pattern with use of MBS can be found in [7].

This way we create a new gait pattern containing reference for feet, pelvis and COM trajectory. This reference is then executed on the robot as described in Sec. II-A

III. STATE ESTIMATION AND FEEDBACK CONTROL

Our research platform COMAN is a platform equipped with Series Elastic Actuators (SEA). In spite of locking the elastic elements for our experiments the joints remain significantly compliant. Even though the gait pattern is generated with use of MBS model, when executed on the robot with elastic components the joint position reference is not tracked properly. This results in real position of COM significantly diverging from the reference. To detect what is the actual position of COM we developed a COM state estimator. Furthermore, to deal with the divergence of COM from the reference trajectory we also developed and implemented a COM feedback controller. In this section we will explain both the state estimator and the COM feedback controller.

A. State estimation

As described above, the joint compliance tracking of position reference is not accurate and thus the state of the robot is different from the reference. To properly estimate the configuration of the robot we use the link level absolute encoders which monitor the position of the robot joints at the link side after the elastic elements. Also since we cannot assume that the support foot is all the time in the flat contact with the ground and that the ground is horizontal, we need to estimate what is an orientation of the whole robot with respect to the ground. For that we use an IMU (MEMS) which is mounted on the pelvis of the robot (Fig. 1). To calculate the COM position we use our MBS model (Fig. 1) comprising 7 masses. Based on the orientation of the pelvis we calculate the position of COM with respect to pelvis. Then we calculate what is the position of COM with respect to ankle joint of the stance foot and finally estimate the COM position in the world coordinate frame by adding the present reference of the stance foot ankle position. To calculate the velocity of the COM we calculate the derivative of position and apply low pass averaging filter with a 30 samples window. The above calculation is performed at every sample of 1 kHz control loop.

The COM state estimator described above was proved to be sufficient for both use with the feedback controller and for gait pattern regeneration.

B. COM feedback control

As mentioned at the beginning of this section, because of the joint compliance of our platform the COM trajectory tracking has significant errors and does not allow for direct execution of gait pattern on the robot. To bring the actual COM position to the reference we use a two stage feedback controller (Fig. 4). The first stage based on the COM position error modifies the ZMP reference in order to bring the COM back to the reference. The second stage is a ZMP controller which is tracking the modified ZMP reference by modifying the pelvis position reference. Since direct modification of the COM could result in ZMP moving to the edge of the support polygon and thus destabilizing the robot, we modify ZMP reference and then modify the pelvis position to follow the modified ZMP (a similar approach was first proposed in [11] without the COM feedback). This approach proved to be more stable than direct modification of pelvis reference. In the first stage of the controller we use a PD controller with gains equal to $K_p^{COM} = 3$ and $K_d^{COM} = 0.4$. In the second stage we used an integral controller with gain $K_i^{ZMP} = 0.003$ and integrate the result to produce the shift in pelvis position. The gains were selected experimentally and proved to be stable in various scenarios throughout the experiments.

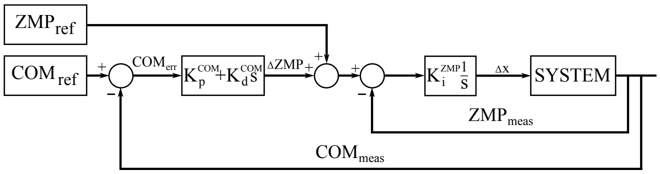

Fig. 4. Block diagram of COM feedback controller. The COM and ZMP references are taken from gait pattern, while measured values are estimated from F/T, IMU and robot proprioceptive (encoders) sensors.

IV. EXPERIMENTS

To prove the validity of the proposed methods we performed a series of experiments on our research platform COMAN (Fig. 1) [8]. It is a medium sized humanoid robot with SEA actuation. It weights 34 kg and is 0.95 m high. As our locomotion controller provides the position reference we partially locked the elastic elements of the lower body actuators. Locking of the elastic elements does not completely eliminate the compliance of the joints, thus the robot should not be compared to the regular position controlled humanoid robots.

We performed three sets of experiments. The first experiment demonstrates the performance of the COM feedback controller described in Sec. III. The remaining two experiments show the effectiveness of gait pattern regenerator (Sec. II) when the robot is subject to lateral push in the direction of the support foot while stepping in place and during forward locomotion. We pushed the robot in the direction of the support foot, as due to the compliance of the structure, a push disturbance in direction of the swing foot results in foot immediately hitting the ground. This can be prevented with appropriate foot position feedback controller, but since this was not the crucial point of our paper we left it for further considerations. We want to note that the gait pattern regenerator has no limitations regarding direction of the disturbance.

A. COM feedback control

In this experiment we compare COM trajectory of the robot with and without COM feedback controller. The robot

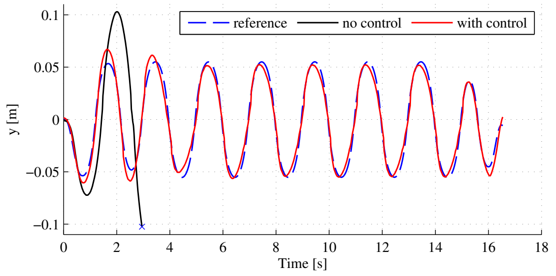

Fig. 5. Comparison of estimated COM trajectory during stepping in place. Without the COM feedback control the overshoot increases with each step eventually resulting in fall at 5th second of motion. With COM feedback controller active the robot is able to complete the stepping motion.

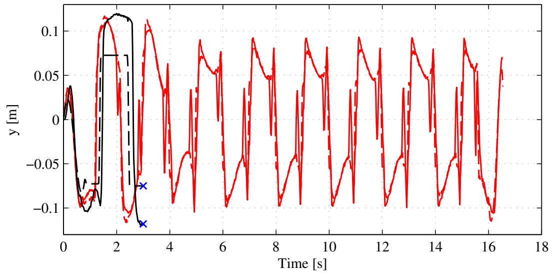

Fig. 6. ZMP trajectory registered during the COM feedback experiment. Red line denotes experiment with control active, while black line denotes experiment with control inactive. The dashed line in both cases denotes the present reference, while solid line denotes measured value.

is executing a pattern in which it is stepping in place. The step time is approximately 1 s and step width is 0.145 m. Since the gait pattern generation process includes MBS model, the resultant motion should include the self-motion (e.g. leg rising) side effects. The model however assumes a perfect position control which is not the case in COMAN. In Fig. 5 we can see trajectory of COM in both cases, with and without the feedback control. We can notice that without the feedback control the COM trajectory keeps diverging eventually leading to a fall approximately during the fifth second of the experiment. The same pattern executed with feedback control switched on results in stable stepping motion with very good COM reference tracking. Fig. 6 introduces the reference and measured ZMP trajectory. It can be observed how the ZMP reference is altered by the controller and how ZMP is tracked by the inner controller.

The effect of the ZMP control to the foot impacts due to an early ground contact was examined demonstrating significant reduction of impacts and smoother step transitions.

B. Trajectory regeneration while stepping

In this experiment we evaluated the response of the robot to a lateral push. The gait pattern is generated as described in Sec. II with the exception that during execution all references in sagittal direction are kept equal to initial value what results in stepping in place. The reference step time and width is the same as in the previous experiment. During the experiment, while stepping, the robot is pushed in lateral direction towards the stance foot. The same experiment was

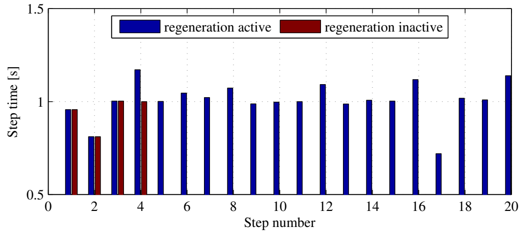

Fig. 7. Step time duration in the second set of experiments.

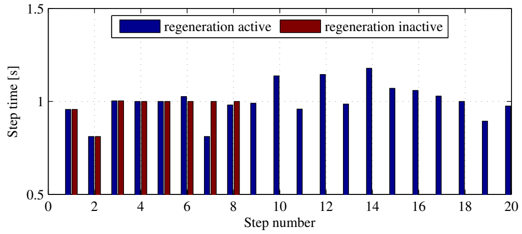

Fig. 8. Step time duration in the third set of experiments.

repeated with the gait pattern regeneration on and off.

Fig. 9 shows the first 14 s of experimental data. The robot with active pattern regeneration completed the stepping successfully (as seen on attached video[1]) and we present only a part of experiment to improve data visibility. In the experiment with gait pattern regeneration disabled the robot was pushed during the fourth step. We can see that it resulted in a large overshoot and even bigger swing of COM in the opposite direction which resulted in the fall of the robot. In the experiment with gait pattern regeneration active, the robot was pushed during the fourth, eighth and twelfth step of the motion (and multiple times afterward, which is not visible on the presented data, but on the accompanying video). After each of the pushes the controller is replanning the step position and step time. In Fig. 7 we see the step time change over the experiment. It is worth mentioning that after a strong lateral push in the direction of stance foot the only way to recover is either to extend a current step and follow the natural swing of a pendulum or to shorten the step time, quickly change the support foot and move the following support foot further away from COM. In both cases there is a need for change in the step time, which this method facilitates. The graph presented in Fig. 9 also shows that the gait pattern regeneration occurred also during tenth step of the motion. In this case the robot was not pushed, but the COM slightly diverged from the pattern and gait pattern regeneration helped to bring it back on track.

C. Trajectory regeneration while walking

In the third and last set of experiments we commanded the robot to walk forward and pushed it in lateral direction. The reference locomotion velocity was set to 0.04 m/s and reference step time was set to 1 s. Results from the first nine steps of the experiment are presented on Fig. 10. Similar

[1] Video accompanying the paper, available online at: http://kormushev.com/goto/IROS-2015-Kryczka

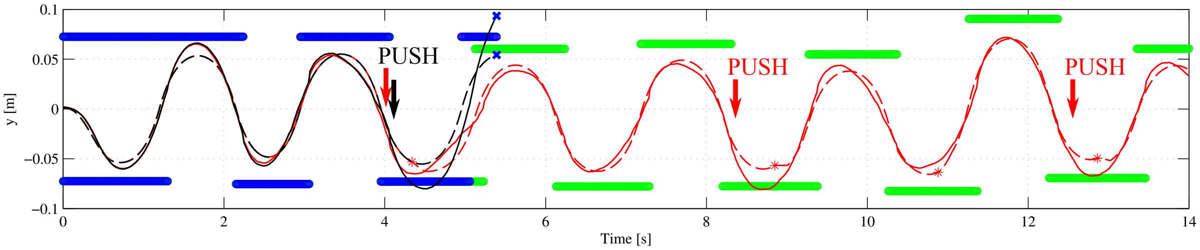

Fig. 9. Graph shows first 14 s of the second experiment in which robot is stepping in place and is being pushed in lateral direction. The thick lines denote ankle position in coronal plane as function of time, continuous lines denote COM measured position while dashed lines denote the COM reference trajectory. Black colored lines correspond to the experiment without gait pattern regeneration, while the red lines correspond to the experiment with gait pattern regeneration active. Red asterisks denote the points at which new pattern was connected.

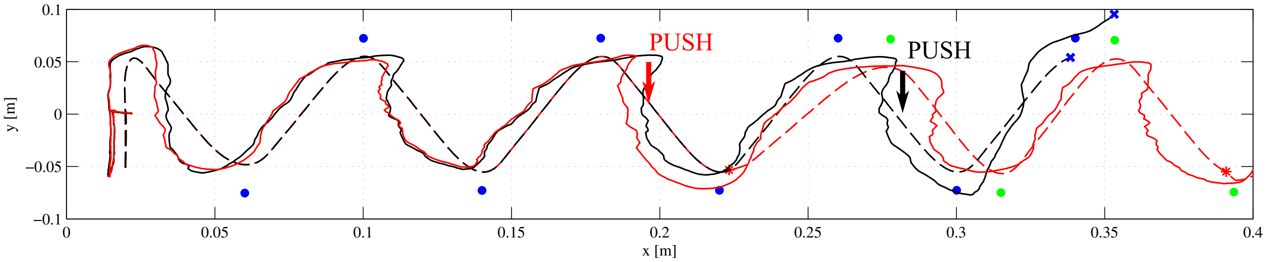

Fig. 10. Graph shows first nine steps of the last experiment in which robot is walking forward and is subjected to lateral push. The meaning of lines and colors is the same as in Fig. 9. In the experiment with pattern regeneration inactive the push occurs during eight step and leads to robot's fall. In the experiment with pattern regeneration active the push occurs during sixth step. One can see that the pattern generator modified the position of step in sagittal and lateral plane.

to previous experiments we truncated the results for better visibility. From the results we can see that the robot with gait pattern regeneration inactive is not able to recover from the push, which occurred in the eighth step. In experiment with gait pattern regeneration active a push occurred during sixth step. From Fig. 10 and Fig. 8 it can be observed that the controller modified step time and step position in order to regain stability.

V. CONCLUSIONS

We presented a method which is capable of gait pattern regeneration and controls not only the step positions, but also the step times to recover from disturbances. In the paper we described the method together with the state estimator and feedback controller used in our locomotion control system. The experiments performed on our robotic platform COMAN showed that the robot is able to recover from strong pushes, even in the direction of the stance foot.

A present formulation of the optimization problem limits application of the method to the horizontal terrain and to locomotion in forward direction, but we believe that this limitation can be overcome and we plan to address it in our future research. Another possibility for extension of the method is to use the whole body angular momentum in the optimization algorithm, which will further increase the range of disturbances from which the robot can recover.

REFERENCES

[1] H. Diedam, D. Dimitrov, P.-B. Wieber, K. Mombaur, and M. Diehl, "Online walking gait generation with adaptive foot positioning through linear model predictive control," in *Proc. IEEE/RSJ Int Intelligent Robots and Systems (IROS) Conf*, 2008.

[2] D. Dimitrov, A. Paolillo, and P.-b. Wieber, "Walking motion generation with online foot position adaptation based on ℓ_1- and ℓ_∞-norm penalty formulations," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 3523–3529.

[3] K. Nishiwaki and S. Kagami, "Frequent walking pattern generation that uses estimated actual posture for robust walking control," in *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, Dec 2009, pp. 535–541.

[4] K. Nishiwaki and S. Kagami, "Trajectory design and control of edge-landing walking of a humanoid for higher adaptability to rough terrain," in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*. IEEE, 2012, pp. 3432–3439.

[5] M. Morisawa, K. Harada, S. Kajita, K. Kaneko, J. Sola, E. Yoshida, N. Mansard, K. Yokoi, and J.-P. Laumond, "Reactive stepping to prevent falling for humanoids," in *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, Dec 2009, pp. 528–534.

[6] J. Urata, K. Nshiwaki, Y. Nakanishi, K. Okada, S. Kagami, and M. Inaba, "Online decision of foot placement using singular lq preview regulation," in *Humanoid Robots (Humanoids), 11th IEEE-RAS International Conference on*. IEEE, 2011, pp. 13–18.

[7] P. Kryczka, Y. M. Shiguematsu, P. Kormushev, K. Hashimoto, H. Lim, and A. Takanishi, "Towards dynamically consistent real-time gait pattern generation for full-size humanoid robots," in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2013.

[8] N. Tsagarakis, S. Morfey, G. Cerda, L. Zhibin, and D. Caldwell, "Compliant humanoid coman: Optimal joint stiffness tuning for modal frequency control," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 673–678.

[9] A. Wachter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[10] 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, pp. 1620–1626.

[11] K. Nagasaka, M. Inaba, and H. Inoue, "Stabilization of dynamic walk on a humanoid using torso position compliance control (in Japanese)," in *Proceedings of 17th Annual Conference of the Robotics Society of Japan*, 1999, pp. 1193 – 1194.