

Online Motion Planning over Uneven Terrain with Walking Primitives and Regression

Sotiris Apostolopoulos^{1,2}, Marion Leibold¹ and Martin Buss^{1,2}

Abstract—This paper introduces an online motion planning algorithm and a motion generation methodology for underactuated dynamic planar walking on uneven terrain. The key idea is to utilize a database of Motion Primitives and use them as training examples in a regression methodology, which is utilized when there is no match between the terrain variation and the Motion Primitives in the database. Among the key features which enable the algorithm to be suitable for real-time purposes is the proposed best first graph search approach and the small inference time of the regression methodology, which in this paper is the Gaussian Process.

I. INTRODUCTION

Underactuated walking robots are trading off control authority with more efficient and human-like locomotion. In order to achieve these objectives, careful understanding of the dynamics underlying their motion is necessary. By that, the natural dynamics of such a robot can be exploited and utilized in the synthesis of sophisticated feedback controllers, which in turn are able to cope with the aforementioned reduced control authority.

However, the dynamics of such robots are highly nonlinear due to the dynamic coupling of the rotational degrees of freedom. Thus, designing walking motions online is computationally costly. As an alternative, a repertoire of walking primitives can be computed offline and employed online in order to steer the motion of the robot based on different terrain specifications.

The idea of utilizing a database of motion primitives in order to control the motion of a robot has been studied in the literature for different kinds of tasks like balancing and motion planning for an aerial vehicle [1], [2]. In the field of walking robots, an early work can be found in [3], where a database of walking primitives was employed in order to navigate JOHNNIE, a fully-actuated humanoid robot. In [4], a compass gait robot was steered through an environment with obstacles using only 3 asymptotically stable walking primitives. Recently, it was shown how a database of motion primitives can be used for motion planning on uneven terrain for a compass gait and a 5-link walking robot [5]. In that work however, the authors assume that the feedback terms

of the controller can compensate for an error between the encountered terrain and the final stride height of the chosen primitive.

Depending on the size of the database and the knowledge of the terrain, different motion planning techniques can be used. For cases with full knowledge of the environment, classic graph search algorithms can be utilized, like in [3], [4]. If a large database renders graph search intractable, online predictive approaches can be utilized so that the search horizon is limited to a few steps and new plans can be generated in each step very fast. The challenge though is to know which sequences of motion primitives are feasible, since in underactuated walking robots there is always the risk that the robot does not have enough kinetic energy to overcome the potential energy barrier. Concrete guaranties whether a sequence of such primitives is feasible or not are given in [6]. In addition, worth mentioning is the work on LQR-trees [7], where reachable sequences of controllers can be generated, even though the region of attraction for each controller is estimated in a conservative way using Sum-of-Squares optimization.

Another challenge is that, even with a very large database of motion primitives, there will always be cases in which there is no primitive corresponding to the current terrain height [5]. In this paper we alleviate this limitation. In order to do so, a regression technique for the generation of walking primitives is introduced and utilized when the database does not contain a primitive which corresponds to the specific terrain variation. This is achieved by designing a database of primitives for walking, ascending and descending stairs and then learning a Gaussian Process which can generate new gaits. Thus, we are able to reduce the cardinality of the database, in order to use efficient motion planning techniques and still react to unforeseen terrain variations. An interesting evaluation of different regression methodologies (including the Gaussian Process Regression) with respect to constraint violation for a fully actuated biped can be found in [8].

Another advantage is that the primitives are extracted based on the Hybrid Zero Dynamics framework [9]. Thus the motion planning utilizes a 2-dimensional state representation instead of the full system state which is always higher-dimensional. The robotic model under study is a 5-link biped, which constitutes a minimalistic model capable of walking on uneven terrain, without suffering from foot scuffing like the compass gait. The underactuation is due to the fact that these robots have point feet. Note however that the methodology is generalizable to any robotic system with one degree of underactuation.

¹Chair of Automatic Control Engineering, Technical University of Munich, Theresienstr. 90, 80333 München, Germany, {s.apostolopoulos, marion.leibold, mb}@tum.de

²TUM Institute for Advanced Study, Technical University of Munich, Lichtenbergstrasse 2a, 85748 Garching, Germany

The research leading to these results has received funding partly from the European Research Council under the European Unions Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. [267877] and partly from the Technical University of Munich - Institute for Advanced Study (www.tum-ias.de), funded by the German Excellence Initiative

The structure of this paper is as follows. In section II some insight on underactuated robotic walking is given and in section III an introduction to the Hybrid Zero Dynamics approach is presented. In section IV the design procedure of the database is explained as well as the conditions under which a sequence of primitives is feasible. In section V, the learning process is explained and evaluated. In section VI, the motion planning algorithm is introduced which is based on a best first approach. Section VII includes two sample cases that validate our approach. Section VIII concludes the paper with a general discussion of the proposed methodology and directions for future work.

II. UNDERACTUATED WALKING

Underactuated walking is modelled as a hybrid process with two discrete states: the single support and the rigid impact. In order to explain these two phases more thoroughly, the dynamics of underactuated walking are introduced. They are based on the Lagrangian formulation and the assumption of rigid bodies. Here, motion is restricted in the sagittal plane, but the proposed methodology can be extended to the 3D case [10].

A. Single Support

During the single support, the legs of the robot are labelled as "stance" and "swing". The stance leg is pinned on the ground and the swing leg moves forward with an adequate foot clearance in order to become the new stance leg, concluding the single support phase. The state \mathbf{x} of such a robot contains the joint positions \mathbf{q} and the joint velocities $\dot{\mathbf{q}}$, i.e. $\mathbf{x} = [\mathbf{q}^T \ \dot{\mathbf{q}}^T]^T$. Thus, a robot with n degrees of freedom has a $2n$ -dimensional state. Utilizing Lagrangian dynamics, the equations of motion can be expressed as

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{B}\mathbf{u}, \quad (1)$$

where $\mathbf{D}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ is the mass-inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n \times n}$ is the matrix of centrifugal and Coriolis terms, $\mathbf{G}(\mathbf{q}) \in \mathbb{R}^n$ summarizes the gravitational terms, $\mathbf{B} \in \mathbb{R}^{n \times (n-1)}$ is the input matrix and $\mathbf{u} \in \mathbb{R}^{n-1}$ is the vector of generalized torques. For this paper, friction and disturbance forces are assumed to be negligible or can be counteracted by feedback control, as in [5]. The challenge in controlling such a system is due to the fact that the input matrix \mathbf{B} is non-square. Thus, when applying input-output feedback linearization methods, there will be dynamics which are non-observable known as *zero dynamics*. However, they have to be taken into account when designing individual walking gaits but also when concatenating different gaits in a single motion plan. Section III explains the *zero dynamics* and the methodology to design feedback controllers for underactuated walking robots in more detail.

B. Rigid Impact

The rigid impact takes place when the swing leg establishes contact with the ground. The impact is assumed to be inelastic and instantaneous. At the rigid impact, the leg previously pinned on the ground (i.e. the stance leg)

loses contact with the ground and the role of the legs is switched. Additionally, the impact causes a discontinuity on the joint velocities $\dot{\mathbf{q}}$ which can be determined by the impact map Δ and the pre-impact joint velocities $\dot{\mathbf{q}}^-$. Instead of introducing an additional model for the single support with the new stance leg, we relabel (or equivalently transform) the coordinates of the robot. Formally, this can be expressed as

$$\begin{aligned} \mathbf{q}^+ &= \mathbf{R}\mathbf{q}^- \\ \dot{\mathbf{q}}^+ &= \mathbf{R}\Delta(\mathbf{q}^-)\dot{\mathbf{q}}^- = \Delta_s(\mathbf{q}^-)\dot{\mathbf{q}}^- \end{aligned} \quad (2)$$

where the plus and minus superscripts denote the post-impact and pre-impact state of the system respectively and \mathbf{R} is the relabelling matrix.

III. HYBRID ZERO DYNAMICS OF WALKING

Control design for underactuated walking robots has been a topic of extensive investigation in the literature [9], [11], [12]. The main idea is to design a set of virtual holonomic constraints and enforce them by input-output feedback linearization. More specifically, the controller design can be formulated as a tracking control problem where the outputs are defined as

$$h_i(t) = q_i(t) - q_i^d(t), \quad i = 1, \dots, n-1, \quad (3)$$

where q_i^d is the desired trajectory corresponding to the i -th degree of freedom (DoF). The index i runs from 1 to $n-1$, since a desired trajectory cannot be enforced on the underactuated DoF q_n (without loss of generality q_n is the underactuated DoF). In walking, time t can be replaced by a monotonically time-increasing variable $\theta(\mathbf{x})$, which replaces trajectories by paths and we can rewrite $\mathbf{h}(t)$ as $\mathbf{h}(\theta(\mathbf{x})) = \mathbf{h}(\mathbf{x})$. This variable is usually the underactuated coordinate of the robot q_n or a function of it. In order to facilitate the design process of the walking controller, a short introduction to the Hybrid Zero Dynamics of walking is necessary. Connection with the robotic model under study can be found in Fig. 1. For more details on Hybrid Zero Dynamics, the reader is encouraged to refer to [9].

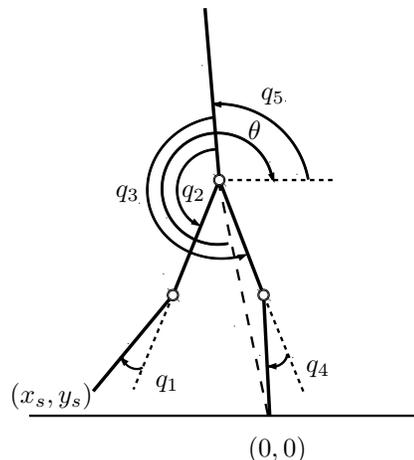


Fig. 1. Kinematic model of the biped under study. The underactuated DoF is the torso angle q_5 . The Cartesian coordinates of the swing leg are denoted as (x_s, y_s) . The x -axis is pointing to the right and the y -axis upwards.

A. Control Law

The main principle in Hybrid Zero Dynamics is the introduction of a coordinate transformation, such that the outputs \mathbf{h} and their time derivatives are zeroed.

Assuming that the dynamics of the robot are expressed in state-space form

$$\begin{aligned}\dot{\mathbf{x}} &= \begin{bmatrix} \mathbf{D}(\mathbf{q})^{-1}(-\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{G}(\mathbf{q}) + \mathbf{B}\mathbf{u}) \\ \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u} \end{bmatrix} \quad (4) \\ &= \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u},\end{aligned}$$

the feedback controller which zeroes the outputs \mathbf{h} is given by

$$\mathbf{u}(\mathbf{x}) = (\mathbf{L}_g \mathbf{L}_f \mathbf{h}(\mathbf{x}))^{-1}(\mathbf{v}(\mathbf{x}) - \mathbf{L}_f^2 \mathbf{h}(\mathbf{x})), \quad (5)$$

where the Lie derivatives are defined as

$$\mathbf{L}_g \mathbf{L}_f \mathbf{h}(\mathbf{x}) = \frac{\partial \mathbf{h}}{\partial \mathbf{q}} \mathbf{D}^{-1} \mathbf{B}, \quad (6)$$

and

$$\mathbf{L}_f^2 \mathbf{h}(\mathbf{x}) = \begin{bmatrix} \frac{\partial}{\partial \mathbf{q}} \left(\frac{\partial \mathbf{h}}{\partial \mathbf{q}} \dot{\mathbf{q}} \right) & \frac{\partial \mathbf{h}}{\partial \mathbf{q}} \end{bmatrix} \begin{bmatrix} \mathbf{D}^{-1}(-\mathbf{C}\dot{\mathbf{q}} - \mathbf{G}) \\ \dot{\mathbf{q}} \end{bmatrix}. \quad (7)$$

Here the arguments of the matrix and vector functions are omitted for brevity. The term $\mathbf{v}(\mathbf{x})$ in (5) is taken to be a PD term

$$\mathbf{v}(\mathbf{x}) = -K_D \mathbf{L}_f \mathbf{h}(\mathbf{x}) - K_P \mathbf{h}(\mathbf{x}), \quad (8)$$

where K_D and K_P are $(n-1) \times (n-1)$ positive definite matrices. Under the control law (5), the outputs are zeroed and the zero dynamics need to be checked for orbital stability.

B. Zero Dynamics Manifold

The Zero Dynamics Manifold is formally defined as

$$\mathcal{Z} = \{\mathbf{x} | \mathbf{h}(\mathbf{x}) = 0, \mathbf{L}_f \mathbf{h}(\mathbf{x}) = 0\}, \quad (9)$$

where $\mathbf{L}_f \mathbf{h}(\mathbf{x}) = \frac{\partial \mathbf{h}}{\partial \mathbf{q}} \dot{\mathbf{q}}$.

Let $\mathbf{x} \in \mathcal{Z}$ and define γ_0 as the last row of the mass-inertia matrix \mathbf{D} , then the coordinates can be transformed into

$$\xi_1 = \theta, \quad \xi_2 = \gamma_0 \dot{\mathbf{q}}. \quad (10)$$

The variable θ is shown in Fig. 1 and is therefore defined as $\theta = \mathbf{c}^T \mathbf{q} = [-1 \ 0 \ -1/2 \ 0 \ -1] \mathbf{q}$. The variable ξ_2 is the angular momentum conjugate to the underactuated DoF q_5 . With this transformation the joint positions and velocities can be reconstructed by

$$\mathbf{q} = \mathbf{H}^{-1} \begin{bmatrix} \mathbf{q}^d \\ \xi_1 \end{bmatrix} \quad \text{and} \quad \dot{\mathbf{q}} = \begin{bmatrix} \frac{\partial \mathbf{h}}{\partial \mathbf{q}} \\ \gamma_0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \xi_2 \end{bmatrix}, \quad (11)$$

where $\mathbf{H} = \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{c}^T \end{bmatrix}$ and $\mathbf{H}_0 = \begin{bmatrix} \mathbf{I}_{n-1} & \mathbf{0}_{(n-1) \times 1} \end{bmatrix}$.

The remaining analysis of the Hybrid Zero Dynamics follows from [9] and is given without any proofs. A difference is made on the description of the fixed point which corresponds to the post-impact state of the robot, instead of the pre-impact one. This difference is done to facilitate the formulation of the Motion Planning Algorithm, as will be described in section VI.

C. Orbital stability of zero dynamics

The derivatives of ξ_1 and ξ_2 can be written as

$$\begin{aligned}\dot{\xi}_1 &= \kappa_1(\xi_1)\xi_2 \\ \dot{\xi}_2 &= \kappa_2(\xi_1)\end{aligned} \quad (12)$$

where

$$\kappa_1(\xi_1) = \frac{\partial \theta}{\partial \mathbf{q}} \begin{bmatrix} \frac{\partial \mathbf{h}}{\partial \mathbf{q}} \\ \gamma_0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (13)$$

$$\kappa_2(\xi_1) = -\frac{\partial V}{\partial q_n} \quad (14)$$

In (14), V is the potential energy function of the system (4). The impact event is taken into account by the resets

$$\xi_1^+ = \theta^+ \quad (15)$$

$$\xi_2^+ = \delta_{\text{zero}} \xi_2^- \quad (16)$$

The quantity δ_{zero} accounts for the angular momentum exchange at the impact and can be computed analytically based on the dynamics of the system, i.e.

$$\delta_{\text{zero}} = \gamma_0(\mathbf{q}^+) \mathbf{\Delta}_s(\mathbf{q}^-) \begin{bmatrix} \frac{\partial \mathbf{h}}{\partial \mathbf{q}}(\mathbf{q}^-) \\ \gamma_0(\mathbf{q}^-) \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (17)$$

Since the zero dynamics manifold is 2-dimensional, the stability analysis is simplified. In this manifold, Lagrangian dynamics can be introduced and kinetic and potential energy functions can be defined as $K_{\text{zero}}(\xi_1)$ and $V_{\text{zero}}(\xi_1)$ respectively. The formal definition of these functions is

$$K_{\text{zero}}(\xi_1) = \frac{1}{2} \left(\frac{\dot{\xi}_1}{\kappa_1(\xi_1)} \right)^2 = \zeta_2 \quad (18)$$

$$V_{\text{zero}}(\xi_1) = -\int_{\theta^+}^{\xi_1} \frac{\kappa_2(\xi_1)}{\kappa_1(\xi_1)} d\xi_1. \quad (19)$$

If the kinetic energy at the beginning of the walking motion ζ_2^+ is greater than the maximum value of the potential energy $V_{\text{zero}}^{\text{MAX}}$, the numeric integration of the zero dynamics (12) will yield a periodic orbit. Formally, if

$$V_{\text{zero}}^{\text{MAX}} - \zeta_2^+ < 0, \quad (20)$$

a periodic orbit exists and if $0 < \delta_{\text{zero}}^2 < 1$, it is exponentially stable. The associated Poincaré Map is given by

$$\rho(\zeta_2^+) = \delta_{\text{zero}}^2 \zeta_2^- = \delta_{\text{zero}}^2 (\zeta_2^+ - V_{\text{zero}}(\theta^-)). \quad (21)$$

The fixed point of this orbit is

$$\zeta_2^* = \frac{\delta_{\text{zero}}^2}{\delta_{\text{zero}}^2 - 1} V_{\text{zero}}(\theta^-), \quad \delta_{\text{zero}}^2 \neq 1 \quad (22)$$

and its domain of attraction is the set

$$D_{\text{zero}} = \{\zeta_2^+ > 0 | \zeta_2^+ - V_{\text{zero}}^{\text{MAX}} > 0\}. \quad (23)$$

Thus, the dimensionality of the system can be reduced from $2n$ to 2. This leads to a 1-dimensional Poincaré Map where the stability analysis can be conducted with analytical expressions. The same holds for the domain of attraction.

IV. MOTION PRIMITIVES

The desired trajectories q^d of the output functions h are parametrized as Bézier polynomials of order M , such that

$$q_i^d(\theta) = \sum_{k=0}^M \alpha_k^i \frac{M!}{k!(M-k)!} s^k (1-s)^{M-k}, \quad (24)$$

where s is defined as $s(q) = \frac{\theta(q) - \theta^-}{\theta^+ - \theta^-}$.

The coefficients α can be obtained through numeric optimization where the relations described in section III regarding feasibility and stability are included in the set of nonlinear constraints. In this paper, these coefficients α are used to define the motion primitives. The optimization process is adopted from [9] and the cost to be minimized is the sum of squared torques per step

$$J = \frac{1}{\ell^-(\alpha)} \int_0^T \|\mathbf{u}(t, \alpha)\|_2^2 dt, \quad (25)$$

where T is the total duration of the gait of the corresponding primitive.

A. Transition between Primitives

A transition between different periodic primitives allows aperiodic walking. Assume that a transition from primitive G_α to a primitive G_β is required. Then the transition is feasible only if

$$\zeta_2^+ - V_{\text{zero}}^{\text{MAX}, \alpha \rightarrow \beta} > 0. \quad (26)$$

If equation (26) is fulfilled then the state of the robot after the impact will be inside the domain of attraction of the periodic primitive G_β . The quantity $V_{\text{zero}}^{\alpha \rightarrow \beta}$ can be computed as in (19) where the integration interval now is from θ_α^+ to θ_β^- . That means, that the joint positions at the end of the transition will be identical to that of primitive G_β , unlike the joint velocities.

B. Database of Motion Primitives

The number of primitives in the database is dependent on the expected terrain variations and possibly on the computational resources available. For the proposed methodology, the cardinality of the database does not need to be large, since the regression technique - as is going to be presented in section V - is able to enrich the capabilities of a small database by generating new primitives when they are needed.

The important quantities in the motion primitives and their explanations are presented in Table I. After defining the role of each term in Table I, a motion primitive P is formally defined as the tuple

$$P = \{\alpha, \theta^-, V_{\text{zero}}, V_{\text{zero}}^{\text{MAX}}, \delta_{\text{zero}}, \ell^-, h^-\} \quad (27)$$

and describes both periodic and aperiodic primitives. In this work, the cost (25) associated with each primitive is not included in the definition (27), since it strongly depends on the post-impact state of the robot x^+ which is not fixed during aperiodic walking.

TABLE I
STRUCTURE OF A MOTION PRIMITIVE

Quantity	Explanation
α	The Bézier coefficients, which describe the feedforward controller (5)
θ^-	Pre-impact value of θ . It is used to define $s(q)$
V_{zero}	Used for the computation of ζ_2^+ during motion planning according to (21)
δ_{zero}	Used for the computation of ζ_2^+ during motion planning according to (21)
$V_{\text{zero}}^{\text{MAX}}$	Necessary in order to evaluate the validity of a transition according to (26)
ℓ^-	Final stride length. Even though it can be reconstructed by the Bézier coefficients α and the value θ^- , it is desirable to be included in the primitive definition in order to facilitate the motion planning and primitive generation
h^-	Final stride height. The justification is the same as for the stride length ℓ^-

V. REGRESSION METHOD

In this section, the regression method is presented, which enables the online generation of walking patterns, when there is no primitive in the database which matches the terrain variation.

For walking on uneven terrain the input is the desired final stride length and height. The outputs are the Bézier coefficients α (see (24)), except the first two, since as shown in [9] they depend on the pre-impact state of the robot x^- , i.e. the previously executed primitive. In this work, the learning model employed is the Gaussian Process, due to its fast learning and inference time and the fact that it was able to learn the nonlinear dependencies between the gaits quite well. A brief introduction to the Gaussian Process follows.

A. Gaussian process [13]

The Gaussian process \mathcal{GP} is a collection of random variables, any finite number of which have a joint Gaussian distribution \mathcal{N} . The Bézier coefficients $\alpha_{2:M}^{1:n-1}$ are generated according to

$$\alpha_{2:M}^{1:n-1}(\phi) \sim \mathcal{GP}(m(\phi), k(\phi, \phi')), \quad (28)$$

where $\phi = [\ell^- \ h^-]^T$ is the input, m the mean function and k the covariance function. The mean and covariance functions can also be defined as

$$m(\phi) = \mathbb{E}[\alpha_{2:M}^{1:n-1}] \quad (29)$$

$$k(\phi, \phi') = \text{cov}[m(\phi), m(\phi')]. \quad (30)$$

The joint Gaussian distribution $\mathcal{N}(\mu, \mathcal{K})$ is defined by a covariance matrix $\mathcal{K}_{ij} = k(\phi_i, \phi_j)$ of dimension $N \times N$, where N is the number of existing primitives and a mean $\mu = [m(\phi_1), m(\phi_2), \dots, m(\phi_N)]^T$. The behaviour of the output functions is determined by the covariance kernels. It is common practice to employ a zero mean function μ and a squared exponential covariance kernel given by

$$k(\phi, \phi') = \sigma_f^2 \exp\left(-\frac{1}{2}(\phi - \phi')^T \Sigma^{-1}(\phi - \phi')\right), \quad (31)$$

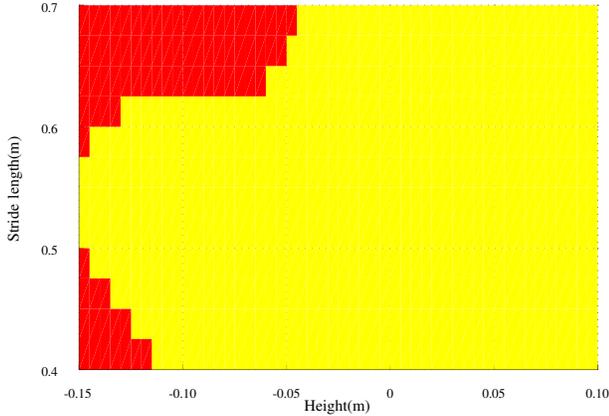


Fig. 2. The range of values for which the regression method can generate periodic gaits. Yellow color indicates periodic gaits, while red indicates non-periodic ones.

where σ_f^2 is the length scale and $\Sigma = \sigma_\phi^2 \mathbf{I}$ the standard deviation with \mathbf{I} the identity matrix. The quantities σ_f and σ_ϕ are referred to as the hyperparameters of the Gaussian Process and are combined in $\lambda = [\sigma_f \ \sigma_\phi]^T$.

B. Evaluation of the Regression Method for Periodic Gaits

The regression method is trained offline and can be utilized online for motion planning. The database of primitives is extracted according to section IV for a grid from 0.4 to 0.7 m for ℓ and from -0.15 to 0.10 m for h . The step increment for ℓ is 0.05 m and for h it is 0.01 m. With these specifications, the database contains 182 periodic primitives as well as all the transitions between them.

The important quantities when a generated gait is employed are δ_{zero} , V_{zero} , $V_{\text{zero}}^{\text{MAX}}$ and the justification is the same as in Table I. The verification of (20) is illustrated in Fig. 2 and the quantity δ_{zero} for periodic walking is presented in Fig. 3 for a densely sampled grid of points corresponding to the stride length and height of the motion primitives in the database. For a gait to be periodic and exponentially stable, (20) as well as $0 < \delta_{\text{zero}}^2 < 1$ must hold. Since, the later holds for the whole grid, the interest is shifted towards Fig. 2. There, it is obvious that the regression method can generate periodic gaits for a very large range of values.

A limitation arises for stair descent with a very large or small step length. At this range, the nonlinear relations between the Bézier coefficients of the periodic gaits are highly uncorrelated and cannot be easily captured by the Gaussian Process. This finding was expected, since during stair descent the robot is taking advantage of gravity and utilizes more of its natural dynamics. Therefore the range of solutions is very small for these gaits. This finding is also supported by the fact that as the height decreases, the range of step lengths corresponding to periodic gaits decreases.

C. Online Generation of Motion Primitives

The value of q_5 at the end of the gait is given by solving the equation $y_s(\alpha, q_5) = h^-$, i.e. equating the vertical component of the swing leg with the desired final stride height (see Fig. 1). This ensures that the generated gaits

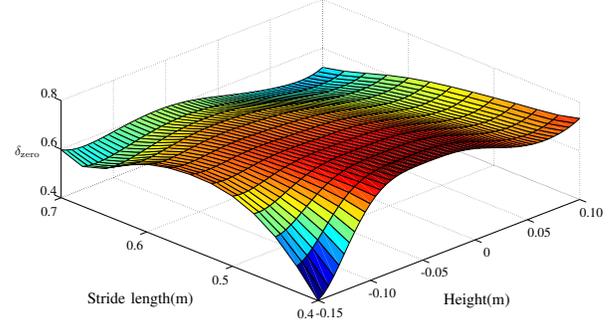


Fig. 3. The quantity δ_{zero} for the generated walking gaits.

will always match the encountered terrain variations. In order to compute V_{zero} and $V_{\text{zero}}^{\text{MAX}}$ very fast, the function V_{zero} is computed on a sufficiently dense grid of points of ξ_1 , so that it can be numerically integrated using the cumulative sum. Since the function V_{zero} corresponds to potential energy, it is concave which makes the computation of $V_{\text{zero}}^{\text{MAX}}$ very easy and fast. The computation of δ_{zero} requires only a simple evaluation of (17). The post-impact joint positions are given by the primitive that was already executed and the pre-impact ones can be extracted by the Bézier coefficients $\alpha_{2:n-1}^{1:M}$ which are the outcome of the Gaussian Process inference.

More precisely, regarding the computational cost, the Gaussian Process regression has an algorithmic complexity of $O(m)$ [14], where m in this case is the total number of training examples. Regarding the dynamic feasibility, the cumulative sum for integrating the function V_{zero} is a linear operation which depends on the size of the grid of ξ_1 points. Finding the maximum $V_{\text{zero}}^{\text{MAX}}$ is also a linear operation.

On the other hand, finding a feasible solution online using optimization constitutes a very difficult problem. The decision variables are $(M-1) \times (n-1)$ and we have nonlinear constraints regarding the feasibility of the system and the periodicity. Even if the optimization is only with respect to the constraints, i.e. finding a set of decision variables that respect the terrain variations and satisfy the constraints, the Gaussian Process is going to return a gait much faster.

Finally, for aperiodic walking an evaluation of the feasibility of transitions is not helpful, since it is subject to the post-impact state of the robot, which is not known beforehand. A solution to this problem is presented in section VI, where the feasibility is checked online with the aid of a look-ahead approach.

VI. MOTION PLANNING ALGORITHM

The motion planning algorithm takes as input the terrain description and gives as an output a sequence of primitives which can traverse this terrain. It is based on a best first approach. This is due to the limited computation time, which is dictated by the duration of the gait. In a few words, we want the motion planning algorithm to terminate before the robot concludes its current step. The sketch of the algorithm is presented in Fig. 4. The search node of the list 'TREE' is assumed to have a structure described in Table II. Initially

TABLE II
STRUCTURE OF A SEARCH NODE

Quantity	Justification
P	Associated primitive as described in table I
ζ_2^+	Necessary for checking the feasibility of transitions and choosing primitives
predecessor	Father of current node in the search tree
k	Depth of the current node in the search tree
(x,y)	Cartesian coordinates of the swing leg on the terrain
ID	Takes discrete values from the set 'N', 'G', denoting nominal and generated node respectively

the structure TREE contains the currently executed node p . At each execution step we choose the best node in TREE. The evaluation criterion assumed for selecting the best node d is defined as

$$d \leftarrow \arg \max_i \{\zeta_{2,i}^+\} \quad (32)$$

The motivation behind this comes from (26). That is, the greater the value of ζ_2^+ is, the greater the number of available primitives gets. The maximum allowed depth of the TREE is D and thus we check if the best node is in this depth.

- If this is true, we backtrack in TREE to find the predecessor g of d in depth 1. If the execution of g will bring the robot past the goal value "terrainX", then the algorithm is terminated. Otherwise, the structure TREE is re-initialized with the node g , whose primitive is executed when the robot concludes its current step.
- If this is not true, we have to check if the node p is a nominal one (p.ID = 'N').

Then we iterate along all the (ℓ^-, h^-) combinations corresponding to primitives in the database to find the ones that match the terrain variations. For that, the ID information of the best node d is important.

- If the best node is a nominal one, we check the dynamic and kinematic feasibility of each primitive P_i .
- If it is a generated one, we utilize the Gaussian Process to generate primitives giving as an input each (ℓ^-, h^-) combination of the primitives in the database and then we check the dynamic and kinematic feasibility of each generated primitive G_i . Generating primitives that match the (ℓ^-, h^-) combinations of the primitives in the database will prevent us from using the regression when it is not necessary, i.e. we have primitives in the database with an (ℓ^-, h^-) combination that matches the terrain variation.

The kinematic feasibility is done by utilizing a set of intermediate configurations and doing collision checks. The dynamic feasibility is checked by computing the quantities V_{zero} and V_{zero}^{MAX} as described in subsection V-C and then evaluating equation (26).

In any case, feasible primitives are appended in TREE. If no primitive in the database was found and no valid primitive was generated, a set of primitives is generated. This procedure is executed for a grid of L equally distributed step lengths ℓ^- in the range $[\ell_{MIN}^-, \ell_{MAX}^-]$. The desired value h^- is equal to the terrain variation in a distance ℓ^-

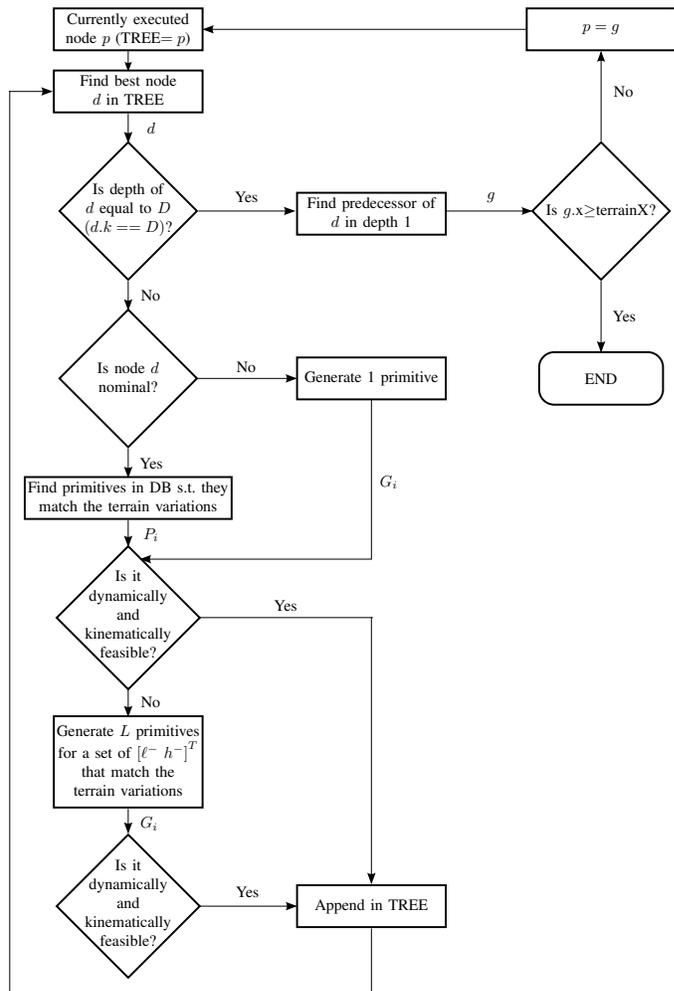


Fig. 4. Flow diagram describing the motion planning algorithm. The kinematic and dynamic feasibility is checked for each primitive that is either chosen from the database or generated from the Gaussian Process.

from the stance leg of the robot. When the L primitives are generated, each is checked again for its kinematic and dynamic feasibility and if it is a valid node, it is appended in TREE. Finally, it should be noted that D and L are design parameters that dependent on the computational resources available.

VII. SIMULATION EVALUATION

This section presents an evaluation of the algorithm proposed in section VI. The robotic model in this experiment matches the parameters of the 5-link biped RABBIT [15].

In Fig. 5 a horizon of $D=3$ steps ahead is utilized and the total amount of primitives which are allowed to be generated by the Gaussian Process is $L=7$. The red configurations denote final robot poses with a $[\ell^- h^-]^T$ specification that does not belong to the aforementioned grid and the regression technique had to be utilized for the transition to and from them. Since the algorithm involves only lower dimensional dynamics and the inference time of the Gaussian Process is small, the proposed methodology can be utilized for online motion planning. An interesting result arises when the second generated gait shows that the robot leans forward in order

VIII. CONCLUSION

This paper proposed an online motion planning algorithm for walking on uneven terrain, using motion primitives which are extracted based on the Hybrid Zero Dynamics approach. The key idea is that, when there is a mismatch between the final stride length of the motion primitives in the database and the terrain height, a regression technique can be used to generate a primitive that matches the terrain variation. For that, the regression method uses the motion primitives in the database as training examples. The algorithm is shown to be efficient since it uses the Hybrid Zero Dynamics of the robot which are 2-dimensional and the inference time of the regression technique (Gaussian Process in this paper) is very small. Simulation studies are presented in section VII. In the future, a more extensive study on the computational advantage of the proposed methodology will be performed.

REFERENCES

- [1] C. Liu and C. Atkeson, "Standing balance control using a trajectory library," in *IEEE International Conference on Intelligent Robots and Systems*, 2009, pp. 3031–3036.
- [2] E. Frazzoli, M. Dahleh, and E. Feron, "Maneuver-based motion planning for nonlinear systems with symmetries," *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1077–1091, 2005.
- [3] J. Denk and G. Schmidt, "Synthesis of walking primitive databases for biped robots in 3D-environments," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2003, vol. 1, 2003, pp. 1343–1349.
- [4] R. Gregg, A. Tilton, S. Candido, T. Bretl, and M. Spong, "Control and Planning of 3D Dynamic Walking With Asymptotically Stable Gait Primitives," *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1415–1423, 2012.
- [5] I. Manchester and J. Umenberger, "Real-time planning with primitives for dynamic walking over uneven terrain," in *IEEE International Conference on Robotics and Automation*, 2014, pp. 4639–4646.
- [6] T. Yang, E. Westervelt, and A. Serrani, "A framework for the control of stable aperiodic walking in underactuated planar bipeds," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 4661–4666.
- [7] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts, "LQR-trees: Feedback Motion Planning via Sums-of-Squares Verification," *The International Journal of Robotics Research*, 2010.
- [8] A. Werner, D. Trautmann, D. Lee, and R. Lampariello, "Generalization of optimal motion trajectories for bipedal walking," in *IEEE International Conference on Intelligent Robots and Systems*, 2015, Sept 2015, pp. 1571–1577.
- [9] E. Westervelt, J. Grizzle, and D. Koditschek, "Hybrid zero dynamics of planar biped walkers," *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 42–56, 2003.
- [10] B. Buss, A. Ramezani, K. Hamed, B. Griffin, K. Galloway, and J. Grizzle, "Preliminary walking experiments with underactuated 3D bipedal robot MARLO," in *IEEE International Conference on Intelligent Robots and Systems*, Sept 2014, pp. 2529–2536.
- [11] D. Djoudi, C. Chevallereau, and Y. Aoustin, "Optimal reference motions for walking of a biped robot," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005, pp. 2002–2007.
- [12] A. Shiriaev, J. Perram, and C. Canudas-de Wit, "Constructive tool for orbital stabilization of underactuated nonlinear systems: Virtual constraints approach," *IEEE Transactions on Automatic Control*, vol. 50, no. 8, pp. 1164–1176, 2005.
- [13] C. E. Rasmussen and H. Nickisch, "Gaussian processes for machine learning (gpml) toolbox," *J. Mach. Learn. Res.*, vol. 11, pp. 3011–3015, 2010.
- [14] D. Nguyen-Tuong, M. Seeger, and J. Peters, "Model learning with local gaussian process regression," *Advanced Robotics*, vol. 23, no. 15, pp. 2015–2034, 2009.
- [15] C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E. Westervelt, C. Canudas-de Wit, and J. Grizzle, "RABBIT: a testbed for advanced control theory," *IEEE Control Systems Magazine*, pp. 57–79, 2003.

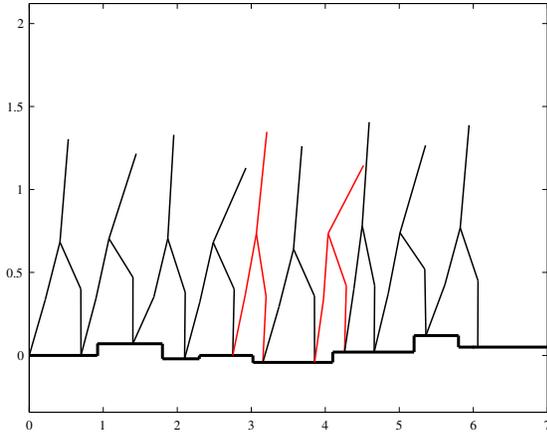


Fig. 5. Walking sequence produced with the algorithm proposed in section VI. The black configurations correspond to gaits that match the aforementioned grid, while the red ones do not.

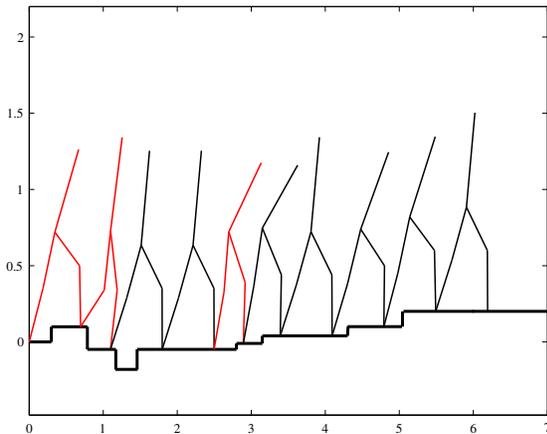


Fig. 6. Walking sequence for a terrain with a challenging transition at the beginning. The Gaussian Process has to be employed to take a transition from a step with a height of 0.099 m to another one with a height of -0.148 m.

to gain more momentum and overtake a big step, as it is indicated on the second nominal primitive as well.

In Fig. 6 another evaluation is presented for a more challenging terrain. For this case, the robot has to take an initial step with a height equal to 0.099 m followed by a step of height equal to -0.148 m. The step down generated transition (2nd red configuration) has a final stride length ℓ^- of 0.4039 m and a final stride height h^- of -0.148 m. According to figure Fig. 2 a periodic gait with this specification cannot be generated. The purpose of this example is to show that even though the generation of such a periodic gait is infeasible, a feasible transition that ends up in a pose with the specified ℓ^- and h^- values can be generated.