Phase Space Planning and Robust Control for Data-Driven Locomotion Behaviors

Y. Zhao, D.H. Kim, B. Fernandez and L. Sentis

***Abstract*—We utilize here regression tools to plan dynamic locomotion in the Phase Space of the robot's center of mass behavior and state feedback controllers to accomplish the desired plans. In real robotic systems, simplified locomotion models and disturbances in the control processes result in deviations from the actual closed loop dynamics with respect to the desired locomotion trajectories. To tackle these challenges, we propose here the use of two control strategies: (1) support vector regression to approximate complex nonlinear center of mass dynamics and plan the feet contact transitions, and (2) sliding mode control to track feet trajectories given the contact timing and location plans. First, support vector regression is utilized to learn a data set obtained through numerical simulation, providing an analytical approximation of the center of mass behavior. To approximate Phase Plane curves, which are characterized by vertical tangents and loop or cyclic behaviors, we use implicit functions for regression as opposed to explicit methods. Based on the proposed regression approximations of the dynamics, we develop contact transition plans and apply robust controllers to converge to the desired feet trajectories. In particular, state feedback controllers might be more convenient than time based controllers in terms of robustness to disturbances. Overall, our methods are capable of learning complex center of mass trajectories and might benefit from the use of robust control techniques. Various case studies are analyzed to validate the effectiveness of the methods including single and multi step planning in a numerical simulation, and swing leg trajectory control on our Hume bipedal robot.**

I. Introduction

An important goal in locomotion is to develop models and control policies that can tackle the difficulty of moving in rough terrains. Data driven models are an attractive solution since they can capture the robot's center of mass nonlinear behavior in the complex terrains. Simple regression models such as polynomial fitting are often used [1], but suffer from conditioning problems, trajectory over-fitting and lack of support to fit curves with infinite slopes or cyclic loops. Among various data-driven fitting methods, support vector regression (SVR) is one of the most interesting approaches and will be leveraged here for locomotion planning and control. SVR is derived from support vector machines, which was first developed to solve classification problems [2], and later extended to solve regression problems [3].

The basic goal of SVRs is to map data sets onto a higher dimensional space via nonlinear mapping. In our study, we employ SVRs to robustly plan biped locomotion behaviors given simulated data sets of the robot's center of mass behavior in the Phase Space. The output of our training model is defined as an implicit surface function that allows to represent the complex center of mass trajectories. The Phase Space is an ideal coordinate system to study stability and robustness of the locomotion behaviors in the rough terrains. As we will soon see, it will allow us to derive stable contact transition policies for switching the locomotion behaviors associated with the hybrid dynamics of the biped.

In the second part of the paper, we will develop feed-forward controllers based on the planned trajectories, and robust feedback servos to provide robustness to disturbances. In particular, sliding mode control will be used to enable convergence to the desired feet trajectories. Given our long term focus on acceleration controllers (i.e., Whole-Body Compliant Control [4]), the State Space motion plans derived here are easily transformed into acceleration control policies and then implemented into hour Hume bipedal robot. The validity of our methods is tested in various simplified dynamic simulations and in an experimental setup involving swing leg trajectory control using our bipedal robot Hume.

Motion planning in robotics has been a focus of attention since obstacle-free mobility became a research topic. We will not attempt here to cover the full scope of motion planning, but a good reference can be found at [5]. Early works on motion planning focused on graph based search, but didn't scale well to highly articulated robotic systems. To solve this issue, randomized search methods were proposed, such as rapidly-exploring randomized trees [6], or probabilistic roadmaps [7], among others. However, these methods only consider the robot's configuration space, but not its dynamic behavior. As a result, extensions to kinodynamic planning were later proposed, e.g. [8], [9]. One problem on these early methods is that motor control policies were separated from the motion plans preventing robustness and realtime responsiveness. Researchers in locomotion have addressed this problem in the form of preview controllers [10], although much of the work has been limited to linearized versions of the robot's center of mass dynamics. Motion planners that incorporate the feedback controller into the trajectory design have been furthered by [11] among others. Disturbances and uncertainty of the motion plans have been recently studied for realtime control in [12]. Synthesis of complex contact behaviors has been thoroughly studied by [13], though not explicitly focusing on dynamic bipedal locomotion. Learning switching policies has been addressed in [14] in the context of planar locomotion and using model reduction, and more recently in [15] using full joint dynamics but limited to planar gaits.

Overall, the main contribution of this paper is two fold: (1) fitting analytical functions to the nonlinear center of mass

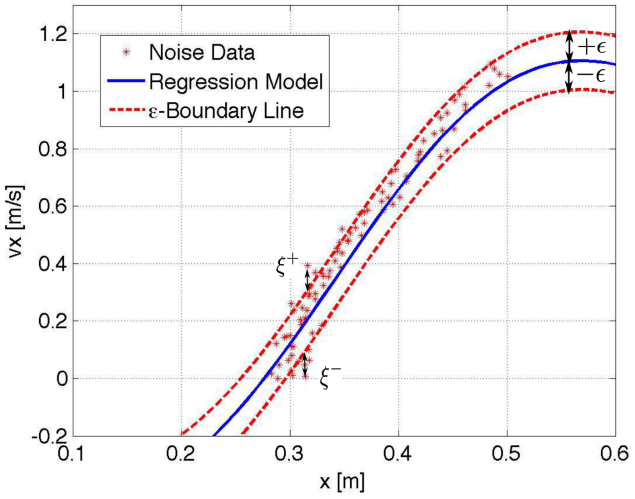


Fig. 1. **Explicit Support Vector Regression of Inverted Pendulum Dynamics performed in the Phase Space.** This figure shows SVR based on a Gaussian kernel (implemented using the software LibSVM [16]). Random noise is added to the position and velocity data (red star dots). The blue line is the nominal learned trajectory after the training. The two red dashed lines surround a ϵ-insensitive tube. Note that some data still exists outside the insensitive tube, which will cause some positive ξ errors. The SVR parameters used here are $C = 2, g = 10, \epsilon = 0.1, b = 0.2937$ (Their meanings will be explained in the optimization formulation.). 11 out of total 88 data sets are selected as support vectors. The main deficiency of this explicit method is that the blue nominal trajectory can not reflect true pendulum dynamics, where the slope around $v_x = 0$ has finite slope. Infinity slope cannot be captured with explicit functions.

Phase Space behaviors using implicit regression; and (2) developing robust controllers for the swing foot that leverage the hybrid trajectory plans learned from the regression process.

II. Background on Support Vector Regression

We introduce a basic overview of support vector regression. SVR is a convenient fitting method that offers higher stability than simple regression methods, enables the adjustment of the fitting sensitivity to noise, and can easily represent high dimensional models.

Suppose that we have a training data set $\{\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n\}$, where $\mathcal{X}_j \in \mathbb{R}^N, j = 1, \ldots, n$ are data pairs, which correspond to physical states such as the position and velocity of the robot's center of mass. The target of SVR is to learn the parameters of a function $\Phi(\omega, b, \mathcal{X})$, expressed as the hyperplane

$$\Phi(\omega, b, \mathcal{X}) \triangleq \langle \omega, \mathcal{X} \rangle + b \tag{1}$$

where $\langle \cdot, \cdot \rangle$ represents the dot product in $\mathbb{R}^N$ and $\omega \in \mathbb{R}^N$ denotes the normal vector to the hyperplane. Note that ω and b are the parameters to be learned. Based on the support vector regression theory [3], these parameters can be obtained by solving the optimization problem

$$\begin{aligned} \min_{\omega, b} \ & \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{l}(\xi_i^+ + \xi_i^-) \\ \text{s.t.} \ & -\epsilon - \xi_i^- \leq \Phi(\omega, b, \mathcal{X}_i) \leq \epsilon + \xi_i^+, \\ & 0 \leq \xi_i^+, \xi_i^-. \end{aligned} \tag{2}$$

where i is the index of i^{th} support vector $\mathcal{X}_i$, l is the total number of support vectors, and ϵ defines the fitting tolerance. Data points are allowed to deviate from the fitting function $\Phi(\omega, b, \mathcal{X})$ with a maximum ϵ value. Additionally, ξ_i^+ and ξ_i^- represent soft margins that allow to reject outliers from the ϵ tolerance tube. Because the number of support vectors decreases with the tolerance ϵ, the larger it is, the smaller the number of support vectors are needed, thus reducing the computational effort. On the other hand, too large ϵ will lead to under-fitting. So choosing the right ϵ value is key to the solution.

It can be shown that the previous optimization problem can be expressed in the following manner (details are omitted)

$$\begin{aligned} \underset{\alpha, \alpha^*}{\text{maximize}} \quad & -\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\langle \mathcal{X}_i, \mathcal{X}_j \rangle \\ & -\epsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{l} y_i(\alpha_i - \alpha_i^*) \\ \text{subject to} \quad & \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0, \alpha_i, \alpha_i^* \in [0, C]. \end{aligned} \tag{3}$$

where α and α^* are Lagrange multipliers representing the first inequality constraint in Equation (2) and the normal vector ω is substituted by the equation $\omega = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)\mathcal{X}_i$. Moreover, Equation (1) can be generalized to

$$\Phi(\omega, b, \mathcal{X}) = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)k(\mathcal{X}_i, \mathcal{X}) + b \tag{4}$$

where $k(\mathcal{X}_i, \mathcal{X})$ is a kernel function. There are several choices for kernels in the SVR literature. In this paper, we will use Gaussian kernels only, i.e.

$$k(\mathcal{X}_i, \mathcal{X}) = exp(-\gamma||\mathcal{X}_i - \mathcal{X}||^2) \tag{5}$$

where γ defines the Gaussian kernel width. Notice that sigmoid and polynomial kernels are also popular for curve fitting. For notation convenience, $\Phi(\mathcal{X})$ will be used for the rest of the paper as a shorthand version of $\Phi(\omega, b, \mathcal{X})$.

III. SVR-Based Motion Planning

We approximate the single contact center of mass behavior of bipeds using nonlinear prismatic inverted pendulum dynamics and then use numerical simulation to obtain data points. Notice that the dynamics of single contact behaviors do not have a closed form trajectory solution in the general case of variable center of mass height. We will tackle the main advantage of using implicit versus explicit functions for regression. First, let us consider an explicit regression problem with two dimensional states.

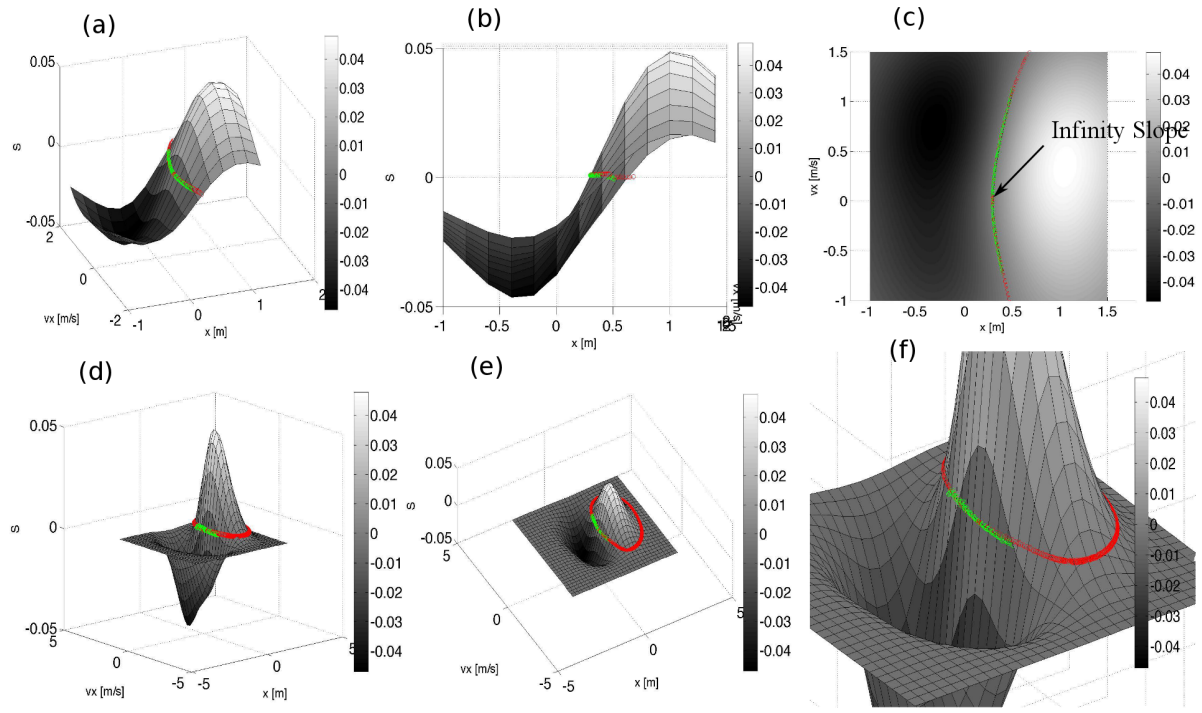


Fig. 2. **Two Dimensional Implicit Regression with Gaussian Kernel:** The SVR parameters are selected as $C = 2, g = 1, \epsilon = 0.02, b = 0.0073$. The number of support vectors is 87. Subfigures (a) - (c) show the fitting surface from different viewpoints. Green is the data points obtained through simulation of a prismatic inverted pendulum while red is the surface fitting. Notice that in (c) we demonstrate that the fitting is accurate even when the slope is infinite on the $(x, \dot{x})$ plane. Subfigures (d) - (f) show the same surface with a longer range of values.

A. Explicit Regression

We assume data points in the Phase Plane represented by a position coordinate, x and a velocity coordinate $\dot{x}$. In explicit regression based on support vectors, a trained model y, expressed by the SVR function

$$y \triangleq \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) k(x_i, x) + b, \tag{6}$$

is solved such that the surface $\phi(x, \dot{x}) \triangleq \dot{x} - y$ approaches zero for all data points with a desired tolerance.

In Fig. 1, we show an example of fitting data points from a nonlinear pendulum simulation using the above explicit function. However, there exist major drawbacks to using explicit functions. First, they cannot represent cyclic loops, such as a circular curves in the Phase Plane. Notice that loop behaviors in the Phase Space are characteristic of locomotion behaviors, representing the periodic cycles. Second, explicit continuous functions cannot, except for a few exceptions, represent vertical tangents (i.e. infinite slopes). For instance, the tangent of the general explicit Gaussian function that we use for regression and given below

$$\frac{dy}{dx} = -2\gamma \sum_{i=1}^{l} (\alpha_i - \alpha_i^*)(x - x_i) exp(-\gamma (x - x_i)^2) \tag{7}$$

can never be infinity since it is the sum of finite positive numbers. However, one of the properties of pendulum dynamics is having infinite slope when approaching the zero velocity axis, e.g. when stopping locomotion at a given step. Given those limitations, we propose instead the use of implicit functions for regression.

B. Implicit Regression

In the case of implicit regression, the fitting function has the expression

$$\Phi(\mathcal{X}) = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) k(\mathcal{X}_i, \mathcal{X}) + b \tag{8}$$

where the implicit state is $\mathcal{X} = (x, \dot{x})$ and $\Phi(\mathcal{X})$ is a continuous differentiable function that defines the sliding manifold. Here, it can be seen that compared with explicit regression, the velocity is coupled with the position state. The partial derivatives are now

$$\frac{\partial \Phi(\mathcal{X})}{\partial x} = -2\gamma \sum_{i=1}^{l} (\alpha_i - \alpha_i^*)(x - x_i) A_{exp} \tag{9}$$

$$\frac{\partial \Phi(\mathcal{X})}{\partial \dot{x}} = -2\gamma \sum_{i=1}^{l} (\alpha_i - \alpha_i^*)(\dot{x} - \dot{x}_i) A_{exp} \tag{10}$$

where $A_{exp} \triangleq exp(-\gamma((x - x_i)^2 + (\dot{x} - \dot{x}_i)^2))$. For visualization, the output response can be treated as a surface and therefore displayed as a third dimension besides the position and velocity states. In Fig. 2 we use once more

prismatic pendulum dynamics to generate data points and use Equation (8) to fit the surface. We then display the three dimensions from various perspectives. As we can see, we are now able to fit the data even if the slope in the Phase Plane is infinity at some point. Moreover, the same implicit method could be used to fit loop trajectories typical of locomotion behaviors in the Phase Plane.

Since the two dimensional hyper-surface $\Phi(\mathcal{X})$ is a differentiable manifold in 3D space, the surface normal can be represented by the gradient

$$\mathcal{N} \triangleq \left[\frac{\partial \Phi}{\partial x}, \frac{\partial \Phi}{\partial \dot{x}}, 1\right]^T \tag{11}$$

Using Equations (9) and (10) and projecting the normal vector in the horizontal plane yields

$$\frac{d\dot{x}}{dx} = \frac{\partial \Phi / \partial x}{\partial \Phi / \partial \dot{x}} = \frac{\sum_{i=1}^{l} (\alpha_i - \alpha_i^*)(x - x_i) A_{exp}}{\sum_{i=1}^{l} (\alpha_i - \alpha_i^*)(\dot{x} - \dot{x}_i) A_{exp}} \tag{12}$$

when $\dot{x} \to 0$ it becomes,

$$\frac{d\dot{x}}{dx} = \frac{\sum_{i=1}^{l} (\alpha_i - \alpha_i^*)(x - x_i) exp(-\gamma((x - x_i)^2 + \dot{x}_i^2))}{-\sum_{i=1}^{l} (\alpha_i - \alpha_i^*)\dot{x}_i exp(-\gamma((x - x_i)^2 + \dot{x}_i^2))} \tag{13}$$

It turns out that the denominator of the above equation becomes very small when approaching the zero velocity axis, thus illustrating that implicit regression can correctly represent vertical tangents, as shown in Fig. 2 (c).

C. Data Generation

The data points shown in Figs. 1 and 2 can be obtained in several ways. In an experimental oriented scenario, the real robot could be controlled to move along predefined geometric paths and starting with various initial conditions. Then data points of the center of mass position and velocity would be recorded for every path. Although this method is an interesting research direction, it is unclear how doable it would be given the large sample space of the center of mass behavior. Instead, for this paper, we will use a simplified model of the dynamics. In the past, we have used a well known nonlinear center of mass dynamic model and applied it to rough terrains [17], represented as

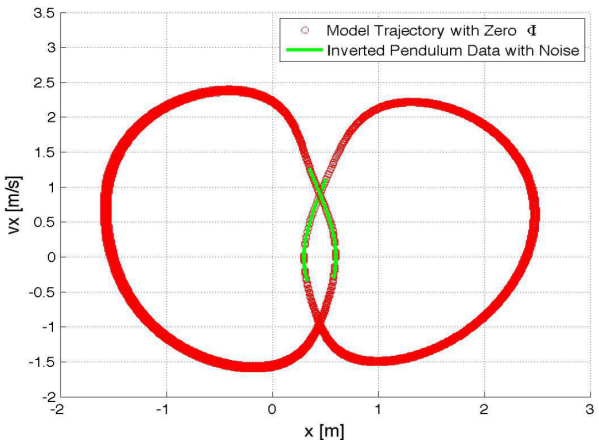
$$\ddot{x} = \frac{(x - p_x)(\ddot{z} + g)}{z - p_z} \tag{14}$$

where x and z are the Sagittal and vertical coordinates of the center of mass and p_x and p_z are the Sagittal and vertical coordinates of the support leg. As it was described in our previous publication about rough terrain locomotion [18], we use geometric primitives to describe the center of mass path above the rough terrain in the form of a nonlinear function, $z = g(x)$. In such case

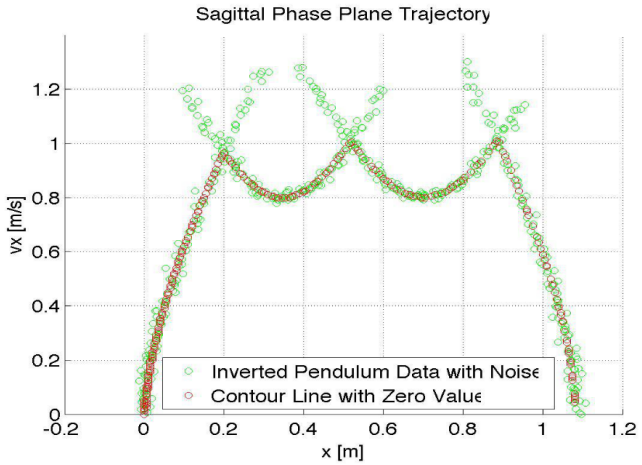
$$\ddot{z} = g''(x)\dot{x}^2 + g'(x)\ddot{x}, \tag{15}$$

which can be used to transform Equation (14) into a nonlinear ordinary differential equation (ODE) with form

$$\ddot{x} = \text{func}(x, \dot{x}). \tag{16}$$



(a) Single Step Planner



(b) Multi Step Planner

Fig. 3. **Contact Transition Planner:** (a) Illustrates the transition point between two adjacent contact dynamics (i.e. two consecutive steps). The upper intersection between the adjacent green color curves represents the contact event that will switch the hybrid dynamics (i.e. the new step). (b) Illustrates the same method applied to multiple steps. In both graphs, the red data is the contour line when the model surface Φ is zero while the green data is the training data used for regression.

Using numerical simulation allows to solve the above equation for the desired geometric path and deliver the data set needed to apply regression.

D. Contact Switching Policy

Once the implicit surfaces for a sequence of contact states have been generated based on the data points, we extract contour lines defined as

$$\mathcal{H}_c(\Phi) \triangleq \left\{\mathcal{X} \mid \Phi(\mathcal{X}) = 0\right\} \tag{17}$$

which give us analytical trajectories to the nonlinear locomotion dynamics. Note that analytical trajectories could not have been directly derived from the nonlinear prismatic pendulum models. The solution for extracting the contact transition points between two adjacent Phase Space trajectories, i.e.

$\Phi_1(\mathcal{X})$, and $\Phi_2(\mathcal{X})$ is obtained by finding the roots of the differential surface

$$F(\mathcal{X}) \triangleq \Phi_1(\mathcal{X}) - \Phi_2(\mathcal{X}) = 0 \tag{18}$$

The solution to the above difference function corresponds to the desired contact transitions. Those transitions will force the robot's center of mass dynamics to move from one trajectory to the next producing the desired locomotion behavior. Single step and multi step examples are shown in Figs. 3 (a) and (b).

IV. Robust Control

The second contribution of this paper, is the design of a robust control approach for locomotion based on the previous hybrid dynamic planner. In the previous section, we developed tools to design a contact switching policy for stable locomotion. And to illustrate the idea, we applied the technique for planing locomotion in the Sagittal plane. In biped locomotion, center of mass Sagittal behavior is not directly controllable due to the dominance of the passive dynamics associated with the small footprint. Instead, it is more convenient to control the swing leg movement so it forces a change of dynamics at the moment where the contact transitions where planned. The idea was presented in [17] but no controller was proposed at the time.

Let us assume for now, that a Phase Space plan has been designed given predefined foot positions and step apex conditions (i.e. the velocity of the center of mass when passing the contact foot), according to the methods described in the previous section. Given the planned contact transitions, we subsequently extract contact timing events from the Phase Space plan. Moreover, once the contact timings are known, time-based foot trajectories can be easily designed. An example of a time based trajectory to swing to a desired contact position is shown in Fig. 4. A valid control approach would be to implement a linear time-based controller, e.g. a proportional-derivative controller based on the trajectory error. However, we consider here the addition of a feedforward term and a robust control term to enhance the tracking performance.

In particular, we consider sliding mode control (SMC) for robustness [19]. Consider the implicit surface $s \triangleq \Phi(\mathcal{X})$ and its time derivative

$$\begin{aligned}\dot{s} &= \frac{d\Phi(\mathcal{X})}{dt} = \frac{\partial \Phi(\mathcal{X})}{\partial x}\dot{x} + \frac{\partial \Phi(\mathcal{X})}{\partial \dot{x}}\ddot{x} \\ &= -2\gamma \sum_{i=1}^{l} (\alpha_i - \alpha_i^*)[(x - x_i)\dot{x} + (\dot{x} - \dot{x}_i)\ddot{x}]A_{exp} \end{aligned} \tag{19}$$

Notice that the state $\mathcal{X}$ above now corresponds to the foot Cartesian trajectory and not to the robot's center of mass trajectory we had considered for the planning phase. We leverage sliding mode control theory for asymptotic convergence to the desired trajectory, e.g.

$$\dot{s} = -\eta \cdot \tanh(s), \tag{20}$$

where η is a control gain and $\tanh(.)$ is the hyperbolic tangent function. The control form above is standard in the SMC literature and it can be easily used to demonstrate that it steers the control variables toward the desired Phase Space trajectory.

One class of controllers that we advocate for is whole-body compliant control [4]. This type of control structure achieves feedback linearization to render full control of task accelerations, i.e.

$$\ddot{x} = u \tag{21}$$

where x is the variable to be controlled and u is the desired control policy. Observing Equations (19) and (20), we isolate the acceleration term and use it as a feedforward term for the above closed loop dynamics, rendering the robust control law

$$u = \frac{\frac{\eta \cdot \tanh(s)}{2\gamma} - \sum_{i=1}^{l} (\alpha_i - \alpha_i^*)[(x - x_i)\dot{x}]A_{exp}}{\sum_{i=1}^{l} (\alpha_i - \alpha_i^*)(\dot{x} - \dot{x}_i)A_{exp}} \tag{22}$$

This law is composed of two parts,

$$u_r \triangleq \frac{\eta \cdot \tanh(s)}{2\gamma \sum_{i=1}^{l} (\alpha_i - \alpha_i^*)(\dot{x} - \dot{x}_i)A_{exp}} \tag{23}$$

$$u_{eq} \triangleq \frac{-\sum_{i=1}^{l} (\alpha_i - \alpha_i^*)[(x - x_i)\dot{x}]A_{exp}}{\sum_{i=1}^{l} (\alpha_i - \alpha_i^*)(\dot{x} - \dot{x}_i)A_{exp}}. \tag{24}$$

where the term u_r corresponds to the so-called *reaching controller*, which drives the system dynamics to the desired surface (i.e. the feedback controller), and the second term u_{eq} is the *equivalent* controller, which forces the system dynamics to move along the surface (i.e. the feedforward term).

In Fig. 5, we show numerical simulations of the above robust control approach to track desired Phase Plane trajectories of the foot swing. Robustness to initial conditions are shown in Fig. 5 (c)-(d) where an initial condition region is chosen to test the SMC law. To avoid chattering due to numerical integration, we use variable step integration. We also simulate external disturbances via velocity impulses. The results are shown in Fig. 5 (e) - (f).

V. Foot Swing Implementation

Initial experiments on controlling the robot's swinging leg are shown in Figs. 6 and 7. A software implementation of whole-body compliant control as described in [4] has been developed to run the experiments on the biped. The robot is supported with a boom system that allows for vertical and pitch motion of the torso. Multiple task frames are defined to control the height of the torso, its orientation and the Cartesian position of the swing foot. Contact constraints on the support foot are accounted for to solve the whole-body torques. A geometric trajectory to swing the foot up and forward then back to its original position is implemented and converted to the Phase Plane for robust control. The regression process described in Equation (8) and the non-linear controller described in Equation (22) are implemented to track the desired trajectories.

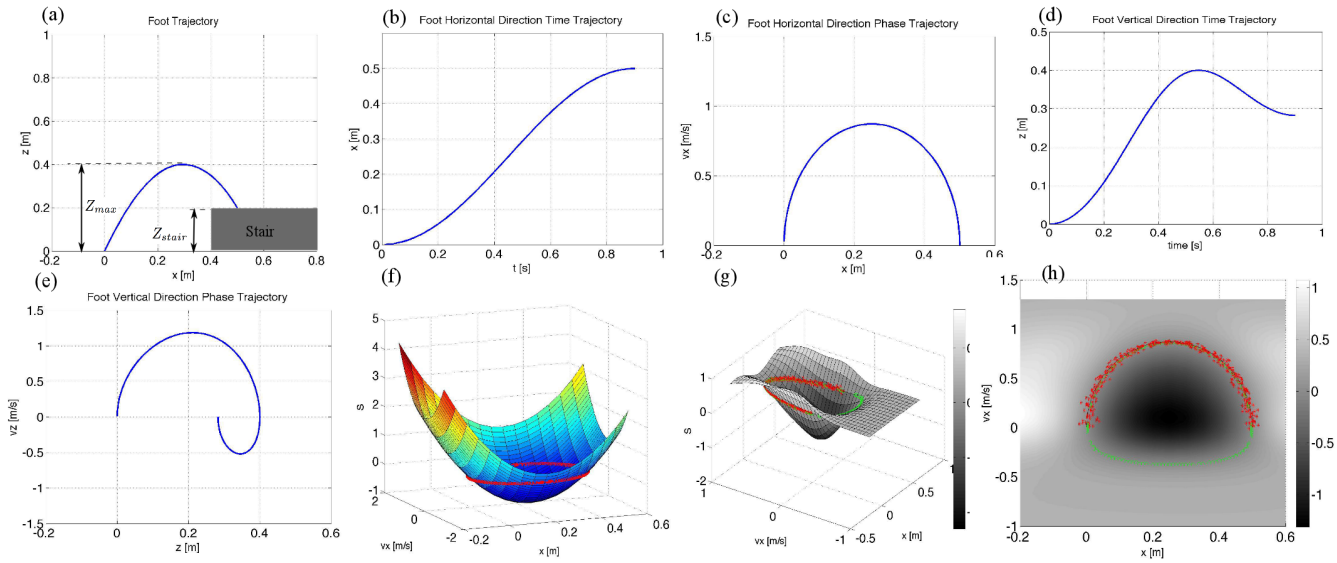


Fig. 4. **SVR-based Foot Trajectory Fitting.** (a) depicts the foot's desired geometric trajectory. (b) and (c) show Sagittal time and phase plane trajectories while (d) and (e) show the corresponding vertical trajectories. In (f), the regression process is shown for those trajectories with the red line corresponding to the zero contour line. (g) and (h) illustrate the Sagittal SVR model from different viewpoints. Notice that in (h), regression captures correctly the infinity slopes near the zero velocity axis.

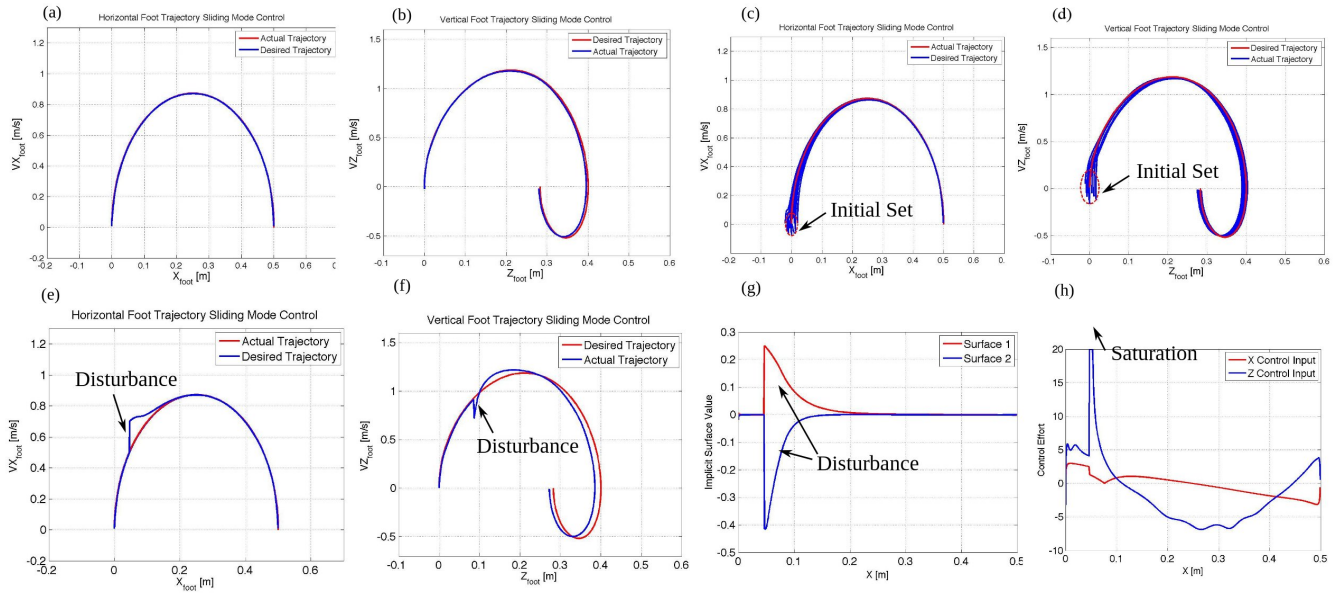


Fig. 5. **Sliding Mode Control Applied to a Trajectory with Various Initial Conditions and Simulated Disturbances:** (a) and (b) show the SMC controller tracking trajectory, undisturbed. (c) and (d) show trajectory tracking under various starting conditions, away from the planned Trajectory. (e) and (f), show the response under a velocity disturbance in both the x and z directions. (g) shows the error surface value for the disturbances applied in (e) and (f). (h) shows the control effort associated with (g).

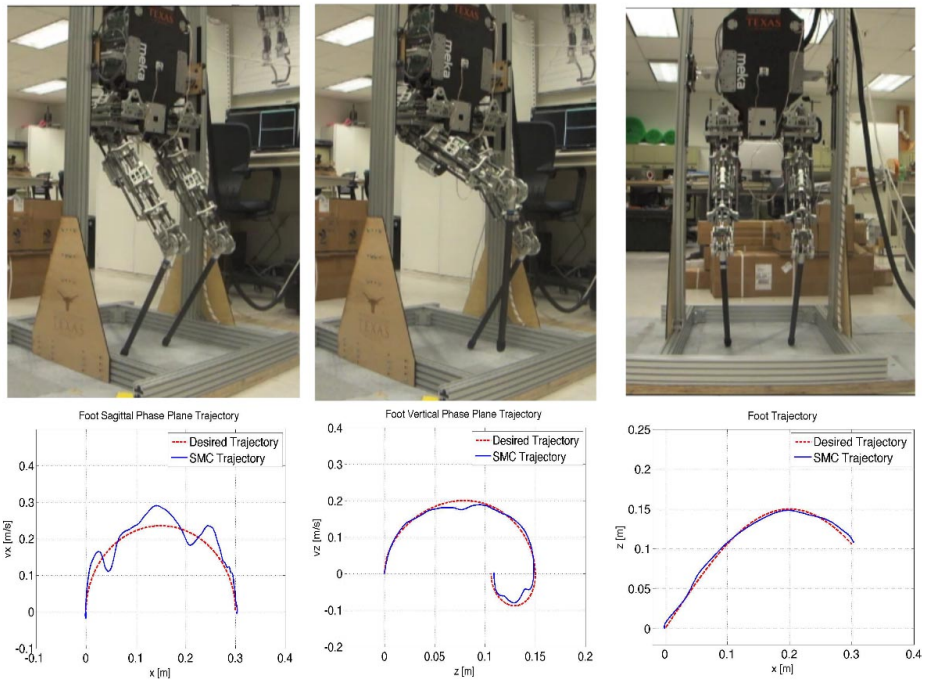


Fig. 6. **Leg Swing Experiment on Hume:** In this sequence we show the implementation of swing leg motions as described in Fig. 4, and based on the implicit regression process with sliding mode control described in Equation (23). The experiment shows the leg swinging to a height, as if it was moving toward a staircase. The controller is effective on tracking the desired path, with accuracy of less than 5 mm.

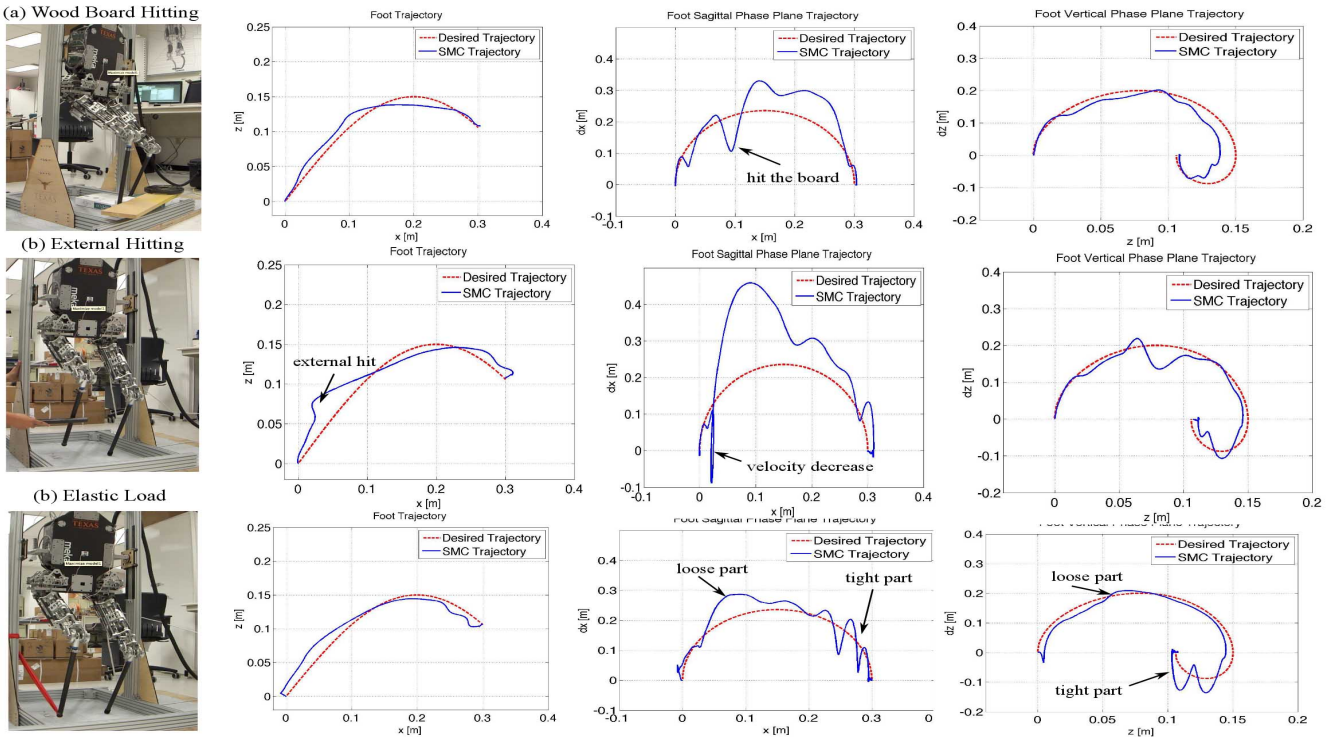


Fig. 7. **Leg swing experiments under Disturbances.** The robust controller is implemented for foot trajectory control. In (a) the leg hits a wooden board which causes sudden velocity reduction. Subsequently the leg pushes the board away to track the nominal path. (b) shows an external force applied to the swing leg, which causes the foot velocity to momentarily reverse direction. In (c), we demonstrate the leg movement when the leg is tied up with an elastic band. As the band tightens up, the leg displays some oscillations.

For further validation of the robustness of our controllers, Fig. 7 narrates three types of external disturbances applied to the swinging leg. The regression based sliding mode control demonstrates satisfactory robust performance. First, Hume's leg hits a wooden board pushing it away in. Second, the leg is hit with a sharp external force. Finally, an elastic band is attached to the robot's leg which tightens up as the leg moves forward.

The experiment suggests that converting the contact transition plans to foot trajectories in the Phase Plane and applying robust controllers for swing leg control could allow bipeds to maneuver and adapt to rugged terrains.

VI. Discussion and Conclusion

Data driven strategies for all terrain locomotion require general models that can determine the contact transitions of the hybrid system. In particular, regression using implicit functions becomes a necessity to fit complex data sets in the Phase Space which normally contain infinite slopes and loop behaviors.

After designing the hybrid control plans, swinging foot trajectories need to be controlled to accurately achieve contact at the desired time and location. Time-based linear controllers are a starting point, but a trajectory based feedforward control policy and robust controller can be key to achieve the needed fast response and robustness to external disturbances. In this context, we have developed a sliding mode control strategy based on Phase Plane plans of a swinging leg and test their effectiveness in simulation and in the Hume bipedal robot.

The two main contributions of our work are utilizing regression tools to deal with nonlinear locomotion models and applying robust controllers for swing leg control based on the contact transition plans. In the future, we will aim at deriving center of mass models from full joint dynamic simulations or using real experimental data from Hume. We also plan to test the locomotion processes during real walking behaviors. To conduct those experiments, a new boom system allowing for Sagittal locomotion is already on its way. Further along, extensions to 3D locomotion of our methods are also on their works.

VII. Acknowledgments

The students involved in this project were funded by the Office of Naval Research. The Hume bipedal robot was funded by The University of Texas at Austin and built by Meka robotics in collaboration with the Human Centered Robotics Lab. We would like to thank all personnel from the Human Centered Robotics Laboratory. Dr. Fernandez leads the NerdLab which is partly sponsored by a grant from the National Science Foundation.

References

[1] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris, *Feedback control of dynamic bipedal robot locomotion*. CRC press Boca Raton, 2007.

[2] V. Vapnik, *The nature of statistical learning theory*. springer, 1999.

[3] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

[4] L. Sentis, J. Park, and O. Khatib, "Compliant control of multi-contact and center of mass behaviors in humanoid robots," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 483–501, June 2010.

[5] J. Latombe, *Robot Motion Planning*. Boston, USA: Kluwer Academic Publishers, 1991.

[6] S. LaValle, "Rapidly-exploring random trees a new tool for path planning," *Report No. TR 98-11, Computer Science Department, Iowa State University.*, 1998.

[7] K. E. Bekris, B. Y. Chen, A. M. Ladd, E. Plaku, and L. E. Kavraki, "Multiple query probabilistic roadmap planning using single query planning primitives," in *2003 IEEE/RJS International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, October 2003, pp. 656–661.

[8] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[9] I. A. Sucan and L. E. Kavraki, "Kinodynamic motion planning by interior-exterior cell exploration," in *Algorithmic Foundation of Robotics VIII (Proceedings of Workshop on the Algorithmic Foundations of Robotics)*, vol. 57. Guanajuato, Mexico: STAR, 2009, pp. 449–464.

[10] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, September 2003, pp. 14–19.

[11] 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*, vol. 29, no. 8, pp. 1038–1052, 2010.

[12] A. Majumdar and R. Tedrake, "Robust online motion planning with regions of finite time invariance," in *Algorithmic Foundations of Robotics X*. Springer, 2013, pp. 543–558.

[13] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behaviors through contact-invariant optimization," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, p. 43, 2012.

[14] J. Rebula, F. Canas, J. Pratt, and A. Goswami, "Learning capture points for humanoid push recovery," in *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*. IEEE, 2007, pp. 65–72.

[15] L. Wang, Z. Liu, C. L. P. Chen, Y. Zhang, S. Lee, and X. Chen, "Energy-efficient svm learning control system for biped walking robots," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 5, pp. 831–837, May 2013.

[16] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[17] Y. Zhao and L. Sentis, "A three dimensional foot placement planner for locomotion in very rough terrains," in *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, 2012.

[18] L. Sentis and M. Slovich, "Motion planning of extreme locomotion maneuvers using multi-contact dynamics and numerical integration," in *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*. IEEE, 2011, pp. 760–767.

[19] V. Utkin, "Variable structure systems with sliding modes," *Automatic Control, IEEE Transactions on*, vol. 22, no. 2, pp. 212–222, 1977.