

# Adaptive Walk-Kick on a Bipedal Robot

Pedro Peña and Ubbo Visser

University of Miami, Coral Gables FL 33146, USA

E-Mail: {pedro|visser}@cs.miami.edu

**Abstract.** Using the NAO robot as a testbed, we propose a walk-kick framework that can generate a kick trajectory with an arbitrary direction without prior input or knowledge of the parameters of the kick in the midst of walking while still guaranteeing reaching a reference trajectory. The walk-kick uses kick interpolators from a dynamic kick engine and the walk trajectories generated from adaptive walking engine to generate motions in any direction that allow a robot to reach its destination while also allowing it to move the ball in further distances without transitioning in different states to accommodate both tasks. The system has been extensively tested on the physical robot, taking into account ten different target angles. The stability and reliability of each kick has been evaluated 30 times for each kick motion trajectory while performing demanding motions. Results show that our proposed walk-kick framework and its integration is reliable in terms of the kick directions and stability of the robot overall ( $< 1\%$  falling rate), and our experiments verify that the walk-kick trajectories were consistent with an average absolute bearing of  $< 6^\circ$  within any given direction.

**Keywords:** Dynamic Kick, Dynamic Walking, Humanoid Robots

## 1 Background and Motivation

Generating dynamic kick motions on a robot while the robot is walking allows the robot to quickly kick the ball towards the goal. This is difficult because the robot can become unstable and might fall as a result. Kick engines such as [2,18,10,17,3,12] have been developed for the NAO robot and although they are dynamic, the kick engines need the robot to be in a stable position where momentum is minimal before they can begin generating kick motions. Although walk-kick controllers such as [19] that use an analytic Zero Moment Point (*ZMP*) have been developed, there are no walk-kick controllers that can generate kick motions in arbitrary directions while the robot is walking. The difficulty of the task lies in generating kick motions that allow the robot to walk while also being in a stable equilibrium.

Wenk et al. [17] developed a kick engine that generates online kick motions using trajectories generated by Bézier curves, but in order to create such motions, the humanoid robot has to dynamically balance on one foot so it can handle any force generated by the kick. In order to find these forces, Wenk et al. use inverse dynamics to calculate the *ZMP*. Böckmann et al. [3] provided a mass spring damper model to model motor behavior, and modified the *ZMP* equation to account for this behavior to get the actual motor position rather than a believed state. The authors

also adapted Dynamic Motion Primitives (*DMP*) to generate kick trajectories, and used a PD controller with a forcing term in the transformation system to control the shape of the trajectory.

Sung et al. [15] use full body motion planning and via-point representation to generate joint angle trajectories. In order to create efficient full body motion trajectories, the authors use optimization techniques such as Semi-Infinite Programming (*SIP*) to specify constraints such as minimal energy and torque, and also deal with joint redundancy. Yi et al. [20] use THOR-OP (*Tactical Hazardous Operations Robot - Open Platform*), a full sized humanoid robot to generate kick and walk motions for the AdultSize League in RoboCup. The robot has a hybrid walking system that uses two types of controllers: a ZMP preview controller and a ZMP based reactive controller. The kick motions generated are handled by the hybrid walking controller to create smooth transitions between the dynamic walk and strong kick.

The kick engine of Xu et al. [18] is separated into four phases: preparation, retraction, execution, and wrap-up phase. The authors use a grid space to find the kick that maximizes the distance in the retraction point and minimizes the angle between the direction of the foot and the direction of the ball. The stabilization of the robot is done with a Body Inclination Control that controls the torso angle to maintain the center of mass (*CoM*) in the support polygon. Becht et al. [2] use a proportional controller to minimize the distance of the *CoM* with regards to the support polygon of the supporting foot, and the kick trajectory is determined by the retraction point and the contact point. Kim et al. [7] use Whole-Body Operational Space Control to generate stable dynamic motions and use a virtual model as an interface for a real robot using an extended Kalman-filter; the controller was tested on a NAO robot generating dynamic kick motions. Lengagneua, Fraisse, and Ramdani [8] use Interval Analysis to find viable motion parameter sets that allow the robot to replan motions based on an offline optimization. A motion optimization for a 30 degrees of freedom humanoid robot, HRP-2, was also done for kick motions where the minimization of energy and joint friction is considered in [9]. Choi et al. [4] developed an impact-based trajectory planner for a kicking robot where the initial velocity and launch angle of the ball are determined by modeling external and aerodynamic forces. Learning kick motions for the NAO robot in simulation has also been investigated [5,16,1].

All of these controllers except the controller from Yi et al. [19] assumes that the robot has no momentum before the kick motion is executed, and there are none that consider dynamic kick motions in any direction while walking. Therefore, we propose a walk-kick controller which integrates a kick controller that generates kick trajectories derived from Peña et al. [12,13] and the walking engine from Seekircher and Visser [14] which makes no assumption of the walk model and allows the robot to use the model as a black box to plan steps while also utilizing a torso angle controller to keep the robot from falling.

In section 2, we will discuss the framework for the walk-kick controller. In section 3, we will describe the experiment design for the walk-kick controller, and the results of the walk-kick trajectories and stabilization will be discussed in section 3.1 and

3.2, respectively. Future directions based on the results of the experiments will be discussed in section 4 and conclusion of the paper will be stated in section 5.

## 2 Approach

Seekircher and Visser [14] developed a dynamic walking engine that uses a 3D linear inverted pendulum model (*LIPM*) to generate a gait. Since there is an analytical solution for the LIPM, it is used to compute the change of the CoM trajectory with respect to the state of the robot which in this case is the foot positions of the robot. The authors optimize the parameters of the step value that satisfies the stable CoM trajectory on the given surface. This relieves the engineer from having to calibrate the robot on different surfaces. The reason for this manual calibration is due to the simplicity of LIPM which does not capture the complete dynamics of the robot such as Coriolis effects from the ground friction forces and nonlinear forces applied to the robot. There are other techniques

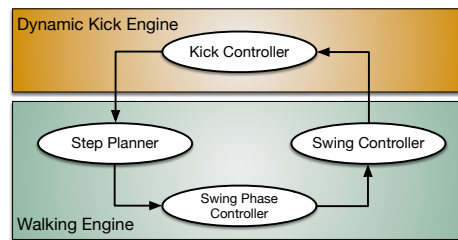


Fig. 1: Walk-kick framework

to find stable motions such as linear and angular momentum control [6] but requires more computation. To generate a kick motion, the robot is required to execute a stable stop motion where the robot has to decrease linear momentum and both foot positions need to be in an initial configuration. Henceforth, it is inefficient and expensive to generate a kick motion unless a kick can be executed while the robot is walking because it will allow the robot to use its linear momentum and generate a kick motion without disrupting the original reference trajectory.

### 2.1 Integration of Walking and Kick Engine

The kick controller resides in the dynamic kick engine [11,12] developed by Peña et al. When the agent sends a request for a walk-kick, the motion request, e.g., kick direction and kicking foot, is received by the step planner. The step planner chooses the next steps of the gait cycle that minimizes the amount of time it takes to reach the reference trajectory dictated by the model which in this case is LIPM. Therefore if the agent requests a kick while the robot is walking, the step planner has to plan for the kick motion while guaranteeing a stable walk. In order to do this, the step planner waits until the current foot for the kick motion is in the swinging phase. When this is the case, the step planner sends a kick direction request to the swing control for a kick motion trajectory. The step planner sends this information to the swing phase controller which keeps track of the current swing foot, and when the swing foot phase of the kick starts, the walking engine communicates with the omnidirectional kick controller from Peña et al. [11,12] to generate a kick trajectory. The walk-kick framework is shown in figure 1.

## 2.2 Walk Trajectory

LIPM is derived from the approximation of a floating-base robot where the robot is approximated by a centroid, i.e., the CoM. The movement of the CoM causes the CoM to tip over because the rest of the body of the robot is approximated by a spoke that is in contact with the ground. Therefore the objective of the floating-base robot is to move its feet such that the inverted pendulum is always balanced and does not tip over. The other assumption made for the LIPM of a floating-base robot is the height is constant. The equation of LIPM is defined as:

$$\mathbf{x}_t = \mathbf{x}_0 \cosh(kt) + \dot{\mathbf{x}}_0 \frac{1}{k} \sinh(kt) \quad \dot{\mathbf{x}}_t = k\mathbf{x}_0 \sinh(kt) + \dot{\mathbf{x}}_0 \cosh(kt)$$

where  $k = \sqrt{\frac{g}{h}}$  and  $[\mathbf{x}_0, \dot{\mathbf{x}}_0]$  are the initial conditions of the system. In order to predict the velocity of the foot in the support exchange, the requested walk velocity and the step frequency can be used to derive the initial seed of the model such as the step target and length. After, Newton's method is used to find a velocity in the first support exchange such that it equals the velocity of the third exchange so the base can walk with constant velocity. In order to maintain the robot balanced, a torso angle controller is used to maintain the robot upright. When the step planner finds a viable foot placement, the walking engine interpolates the feet position to this position.

## 2.3 Walk-Kick Trajectory

The foot positions provided by the step planner are used to generate motions that allow the robot to hit a ball while still following the reference trajectory. In pursuance of a kick motion, the kick controller has to have a notion of the step planning of the robot. The kick motion trajectory is then generated using a trajectory generated from one of the interpolators [13]: sextic, cubic, or Hermite cubic splines. Hence the walk-kick trajectory controller is as follows:

$$\mathbf{x}_{t+1} = \psi(\mathbf{x}_s, \mathbf{x}_f, \mathbf{x}_{via}, t_f, t_{curr}), \quad \dot{\mathbf{x}}_{t+1} = \dot{\psi}(\mathbf{x}_s, \mathbf{x}_f, \mathbf{x}_{via}, t_f, t_{curr})$$

where  $\mathbf{x}_t = [x_t, y_t, z_t]$  and  $\dot{\mathbf{x}}_t = [\dot{x}_t, \dot{y}_t, \dot{z}_t]$ .  $\psi$  is one of the interpolators from [13], and in every time step the swing controller uses  $\psi$  and  $\dot{\psi}$  to calculate the position and velocity of the swinging foot. For the kick motion, two cubic polynomials are generated when the cubic spline interpolator is used. The point where the two polynomials meet is called the via-point. The cubic polynomial is as follows,

$$\alpha_1(t) = a_{13}t^3 + a_{12}t^2 + a_{11}t + a_{10}, \quad \alpha_2(t) = a_{23}t^3 + a_{22}t^2 + a_{21}t + a_{20}.$$

In order to generate an arbitrary motion, specific constraints need to be put upon the polynomials. Since there are two cubic polynomials (i.e., eight coefficients / DOF), there are eight constraints. Every piece of the cubic Hermite spline is a polynomial of three degrees. The cubic Hermite spline is specified by its end points and the tangents of those points (i.e., derivatives at the end points). The cubic

Hermite spline is a smooth continuous function and to interpolate the Hermite polynomial, the following definition is used:

$$\mathbf{p}(t) = (2t^3 - 3t^2 + 1)\mathbf{p}_0 + (t^3 - 2t^2 + t)\mathbf{m}_0 + (-2t^3 + 3t^2)\mathbf{p}_1 + (t^3 - t^2)\mathbf{m}_1.$$

This equation can be used to interpolate the foot of the robot through the kick trajectory. The above interpolator is for unit intervals. Therefore,  $0 \leq t \leq 1$  and  $\mathbf{p}(t)$  is the position of the foot at time  $t$ .  $\mathbf{p}_0$  and  $\mathbf{p}_1$  are the end points of the kick trajectory, and  $\mathbf{m}_0$  and  $\mathbf{m}_1$  are the derivatives of the end points. The velocity of the foot at time  $t$  can be defined as the derivative of  $\mathbf{p}(t)$ :

$$\dot{\mathbf{p}}(t) = (6t^2 - 6t)\mathbf{p}_0 + (3t^2 - 4t + 1)\mathbf{m}_0 + (-6t^2 + 6t)\mathbf{p}_1 + (3t^2 - 2t)\mathbf{m}_1.$$

Using sextic polynomials as a kick interpolator lets us define seven constraints due to the seven coefficients in the polynomial. The sextic polynomial is defined as:

$$\alpha(t) = a_6t^6 + a_5t^5 + a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0.$$

The seven constraints of the kick trajectory are used to control the position, velocity, and acceleration of the kick. It is also used to control the height of the foot from the ground at the via point. There is still a notion of the via point in the sextic polynomial but a knot where two polynomials meet does not exist. The via point is only used to control the swing shape of the kick trajectory. With these seven constraints, the linear system can now be solved to obtain a polynomial that is able to generate the kick trajectory desired.

The starting position (retraction point),  $\mathbf{x}_s$ , of the kick trajectory is the same starting position of the step planner. In fact, the only difference between a regular walk trajectory and a walk-kick trajectory will be the via point,  $\mathbf{x}_{via}$ , in which the via point is the actual contact point of the foot and the ball, and it is what makes the trajectory a kick motion. Lastly, the final position of the swinging leg,  $\mathbf{x}_f$ , is the same as the step planner, and when the swing phase ends, the swing control sends a finish signal to the kick controller. The time constraint,  $t_f$ , for the trajectory is decided by the step planner, and it is not altered so that the step trajectory of the gait cycle is not tampered with allowing the walking engine to seamlessly reach its reference trajectory while also executing kick trajectories to move the ball at further distances.

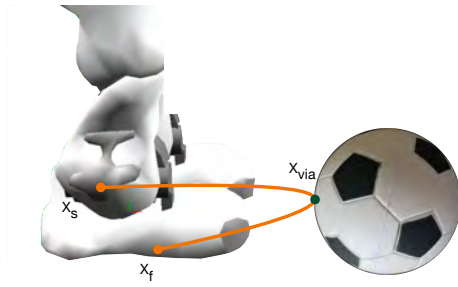


Fig. 2: Walk-kick motion where the via point is the point of contact with the ball.

In order to interpolate the kick trajectory during the swinging phase, the swing controller sends the kick controller, shown in figure 1, the current time,  $t_{curr}$ , to interpolate the spline. While the walk-kick trajectory is executed in the swing phase, the swing controller is communicating with the kick controller for the position  $\mathbf{x}_t$  and velocity  $\dot{\mathbf{x}}_t$  of the foot in the next state. When the swinging foot reaches  $\mathbf{x}_f$  at time  $t_f$ , the kick controller ends. The

kick controller now waits for another request and the cycle starts again. An example of how the walk-kick trajectory is generated is shown in figure 2. The start position,  $\mathbf{x}_s$ , is the first point in the curve from the left. The second point is the via point,  $\mathbf{x}_{via}$ , i.e., the contact point to the ball, and the end position,  $\mathbf{x}_f$ , is the next step position for the support phase. Note that the first and second position are the same positions from the step planner.

### 3 Experimental Design & Results

To test the validity of the integration of both the kick and walk controllers, we designed two experiments that tested the reliability of the kick trajectories and the ball trajectory while also testing the stability of the system while it generates these motions in the course of walking. The first experiment conducted was to test the performance of the kick controller while the robot walks as explained in section 3.1, and in the second experiment, the stability of the robot was tested by observing the CoM trajectory which will be discussed in section 3.2. The experiments were done on a soccer SPL field for ten kick directions: front, back, front-diagonal ( $45^\circ$ ), back-diagonal ( $45^\circ$ ), and side for the right and left foot. The experiments were done with Softbank’s NAOv4 robot. The robot was placed on the field such that the kick direction of the robot was perpendicular to the goal line. For example, for the front kick, the robot was placed facing the goal, while for the side kick, the robot was placed parallel to the goal line. For all kicks, the robot was placed by the half line. On the other hand, the ball was placed near the kicking foot of the robot and the robot walked roughly in place for several steps and kicked. The reason for this was to mitigate the variability of the kick direction as the robot walked up to the ball. When the robot walk-kicks, the controller timed the amount of time it took for the robot to generate the kick trajectory, and the ball trajectory was timed with a stop watch. The x and y distance of the ball trajectory was measured on the field using a laser instrument, Bosch GLM50c. The measurements were taken with respect to the kick direction of the robot. This was done 30 times for each direction and for each foot for a total for 300 kicks; results of the position, bearing, and speed of these kicks are visualized in figure 3. This was done in the course of three days for about  $\sim 40 - \sim 75$  minutes for each direction and foot, so that overheating motors do not become an influencing factor in the experiment. Lastly, the experiments were done on the physical robot and not in simulation.

Table 1: Mean/Std. Statistics: (Left:  $\mu, \sigma$ /Right:  $\mu, \sigma$ ). Normal: (Left: range p-value, bearing p-value/Right: range p-value, bearing p-value), numbers are bold if normally distributed according to the Anderson-Darling test.

Direction (L/R)	Range (cm.)	Bearing (deg.)	Ball Speed (m/s)	Normal, $\alpha = 0.05$
Front	80.0, 22.3/90.5, 19.3	-27.3, 19.1/4.0, 12.0	0.37, 0.1/0.34, 0.1	<b>0.47</b> , 0.04/ <b>0.35</b> , 0.1
Front-Diagonal	97.8, 24.8/99.2, 24.3	0.7, 19.5/-5.7, 15.4	0.55, 0.1/0.59, 0.2	<b>0.32</b> , <b>0.67</b> /0.02, <b>0.15</b>
Side	63.2, 25.1/77.1, 17.6	-5.0, 17.9/0.8, 12.6	0.48, 0.2/0.45, 0.1	<b>0.79</b> , 0.0/ <b>0.73</b> , 0.0
Back-Diagonal	148.1, 21.5/106.5, 22.9	-0.9, 8.8/-10.6, 17.8	0.65, 0.1/0.54, 0.1	<b>0.48</b> , <b>0.36</b> / <b>0.34</b> , <b>0.5</b>
Back	143.5, 50.0/156.4, 47.3	-9.3, 15.0/-3.7, 11.7	0.57, 0.2/0.61, 0.1	0.0, 0.01/ <b>0.05</b> , 0.0

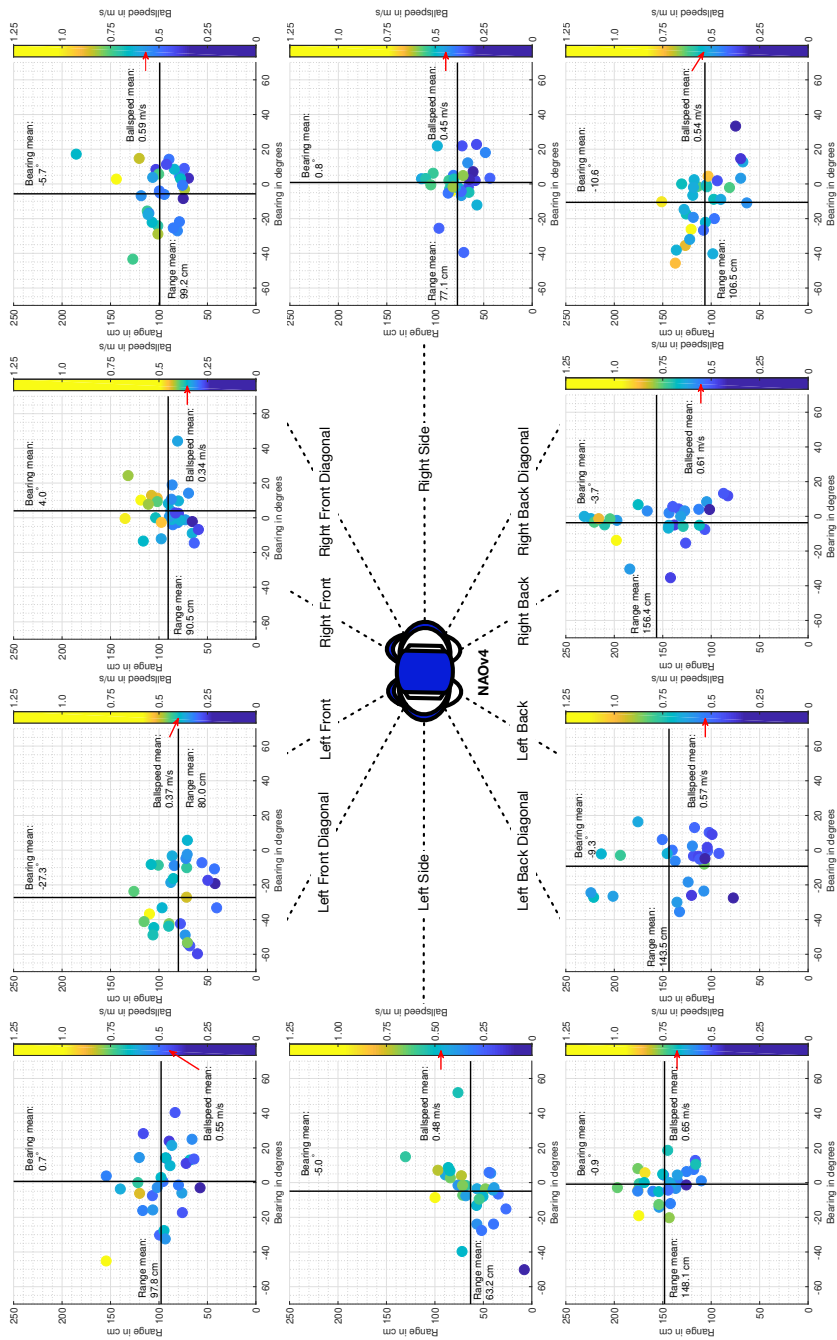


Fig. 3: Walk-kick range and bearing for each kick direction done 30 times with a total of 300 kicks

### 3.1 Kick Trajectory

In general the walk-kick was very stable, regardless of the type of kick direction with only 2 out of 300 kicks that resulted in a fall – one right front kick and one right back kick – which is  $< 1\%$  (0.66%). The mean of the range for all kick directions is 106.2 cm with a minimum of 7.8 cm and a maximum of 259.2 cm. The mean of the bearing for all kick directions is  $-5.7^\circ$  with an interval of  $-59.7^\circ$ -  $129.8^\circ$ . The mean ball speed was 0.52 m/s with a minimum of 0.08 m/s and a maximum of 1.1 m/s. The mean and standard deviations for range, bearing, and speed for each kick direction is summarized in table 1 and can be visualized in figure 3.

Overall the kicks were very stable and the intended ball direction was consistent with an average absolute bearing of  $< 6^\circ$  which shows that the walk-kick is able to kick the ball in any direction relatively accurately while the robot is walking. (cf. figure 3). The first observation we noticed during the experiments is the left front kick was the worst performing kick in terms of kick direction reliability with an absolute bearing error  $> 25^\circ$ , i.e., the ball goes more left diagonal rather than left straight forward.

It is also important to note that both right front kick and right back kick were stronger as can be verified in table 1, i.e., range and bearing, than the left front kick and left back kick; one possible explanation is it has a weaker left leg than the right leg. All back kick directions, e.g., left back diagonal, left back, right back, and right back-diagonal, were the strongest of all with an average range of 138.6 cm compared to an average mean of 84.6 cm for the rest of kick directions, but were less reliable with respect to the kick direction with an average bearing of  $-6.1^\circ$  compared to an average bearing of  $-5.4^\circ$ . A surprising finding was both right front and left front are weaker compared to other directions – mean kick of 85.3 cm, bearing of  $-11.7^\circ$ , and mean speed of 0.36 m/s compared to 111.5 cm,  $-4.2^\circ$ , and 0.6 m/s – because these kick directions are standard kick directions used for experiments in [10,2,18,17,3] and the rest of kick directions have not been explored entirely.

To verify whether there was a correlation between range and bearing, we did a correlation test using the Spearman rank correlation coefficient and found no significant result for range and bearing but found a significant correlation between range and speed with a p-value of 0.0 (the result can be seen in figure 4a); the histograms of the correlation matrix in figure 4a are also approximately normal. Because of the walk-kick variability when the robot walked up to the ball and kicked, it was possible for the ball to take curved trajectories. Although this might have been possible, all measurements were taken as if the ball traveled in a straight trajectory. For this reason, there was variability in the relation between ball speed and range for all kicks, but overall, the statistics in figure 4a show that range and speed were correlated. The normality of the densities were tested with the Anderson-Darling test with an  $\alpha$  of 0.05. Table 1 shows the results of each kick direction density for range and bearing. As can be observed, most of the densities for range with the exception of left back and right front-diagonal are normally distributed. Because the experiment was done from a position where the robot had to walk up to the ball and walk-kick, contact with the ball was inconsistent and most of the kick direction densities for bearing were approximately negatively skewed as can be verified in table 1. The



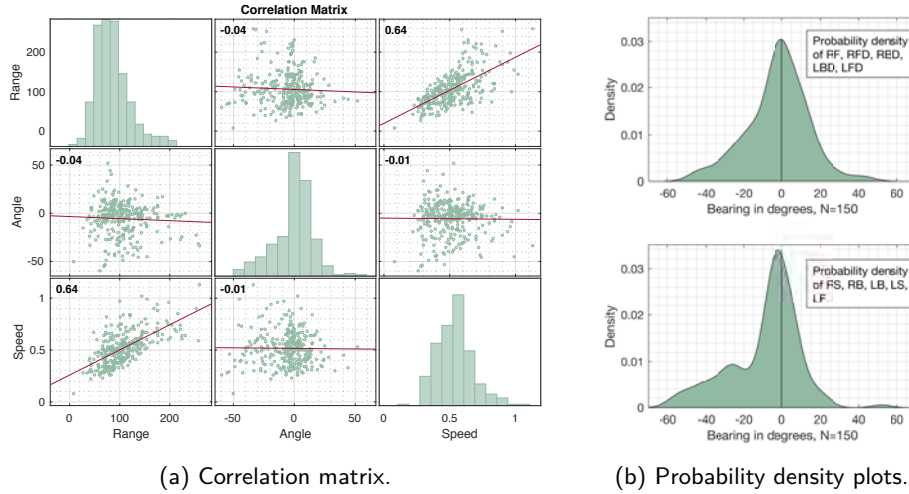


Fig. 4: a) Correlation matrix between range, bearing and ball speed using the Spearman rank correlation coefficient. Only the correlation between range and speed is significant ( $r = 0.64$ ;  $p = 0.0$ ). b) Probability density plots for various kicks Top: Right front (RF), Right Front Diagonal (RDF), Right Back Diagonal (RBD), Left Back Diagonal (LBD), Left Front Diagonal (LFD). Bottom: Right Side (RS), Right Back (RB), Left Side (LS), Left Back (LB).

bottom density of figure 4b shows the density of the bearing for all kick directions that were below the  $\alpha$  value in the Anderson-Darling test, e.g., right side, right back, left back, left side, and left front, where the density is approximately normal with a skewness value of  $-0.62$  which means it is moderately negatively skewed and it is leptokurtic, with a kurtosis of  $3.7$ , compared to the density of the bearing for all the kick directions that were above the  $\alpha$  value in the Anderson-Darling test, e.g., right front, right front-diagonal, right back-diagonal, left back-diagonal, and left front-diagonal, with a skewness value of  $-0.28$ , i.e., approximately symmetric, and kurtosis of  $3.69$  shown on top of figure 4b. Hence, even with random ball contacts, most of the densities of the range in table 1 and half of the densities for the bearing were normally distributed, e.g., top bearing density of figure 4b.

### 3.2 Stability

While the robot is kicking in the swing phase of the gait cycle, the robot can become unstable and fall. Since the walking engine is adaptive, we wanted to stress test the robot while it was walking to see if the robot recovered and reached its reference trajectory. We therefore wanted to record the CoM trajectory while the robot was walk-kicking. In order to do this, we recorded the CoM trajectory of the robot while it was walking and walk-kicking. The CoM trajectories were recorded with respect to the torso of the robot. This data was recorded on a file for all 300 kicks. All

the CoM trajectories were visualized on a graph for each kick and foot, and after plotting all trajectories, the average trajectory was plotted.

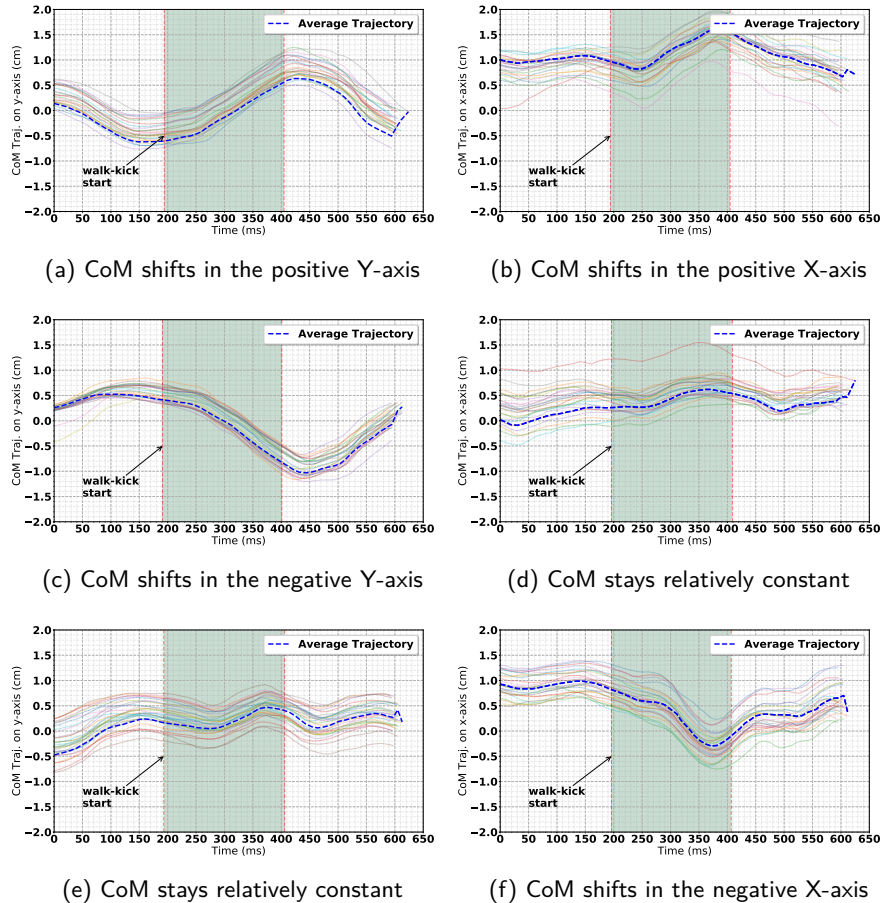


Fig. 5: Walk-Kick CoM Trajectories

As shown in figures 5a-5f, the CoM trajectories follow a similar pattern, and as expected they follow a trajectory that is predictable and expected when the stability of the system is guaranteed. Figures 5a-5f show all the CoM trajectories exhibited by all 300 kicks and show very interesting results. The first observation is that each graph shows the CoM behaving uniquely where the walk-kick region, which is shaded in figures 5a-5f, shows three different behaviors: positive slope, negative slope, and constant.

Each of these behaviors explains a different phenomenon and every combination of the x-axis and y-axis describes a unique kick direction where even the kicking foot

can be derived. Figure 5a is the CoM trajectories for all kick trajectories that moved on the positive y-axis, positive slope, of the robot, e.g., left front-diagonal, left side, and left back-diagonal. As can be seen in the graph, all the y-axis kick trajectories moved as expected: as the robot kicks the CoM shifts on the positive y-axis because of the swinging foot leaves the center of the robot while also showing that CoM stays in the center of the support polygon shown in figure 6a. When the robot kicks in the negative y-axis, negative slope, such as the right side, right front-diagonal, and right back-diagonal, the CoM shifts toward the negative y-axis. When the robot kicks on the x-axis and has a constant y-value, e.g., right and left front kicks and right and left back kicks, the CoM stays relatively close to zero on the y-axis such as can be seen in figure 5e.

The same effects can be seen for kicks that move in the positive x-axis, figure 5b, the negative x-axis, figure 5f, or relatively close to zero such as in the case when the robot does a side kick with either foot as shown in figure 5d. Since all these kick trajectories can be classified with these CoM trajectories, the combination of two CoM trajectories can explain a unique kick direction. For instance, if the robot will execute a left front-diagonal kick, we expect the CoM to shift towards the positive x-axis, positive slope, and y-axis in which figures 5a and 5b explain this phenomenon. The same can be said for any kick direction – e.g., figures 5c and 5f for right back-diagonal kick and figures 5a and 5f for left back-diagonal kick – and this concept was supported in the experiments conducted. An important question that is raised is

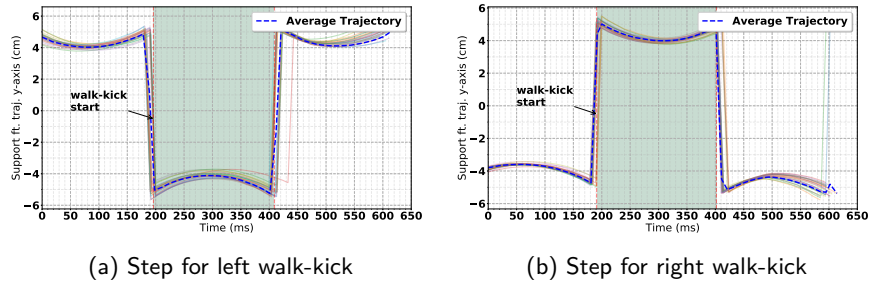


Fig. 6: Walk-Kick Step Trajectories

whether the robot becomes unstable when all these shifts of CoM occur. As can be verified in these graphs, the CoM for the 300 kicks were bounded between -2.0 - 2.0 centimeters, figures 5a-5f, in which the supporting foot trajectories were bounded by -6.0 - 6.0 centimeters, figures 6a and 6b, which shows that the CoM was always in the support polygon of the robot; this verifies why the likelihood of the robot falling when walk-kicking is  $< 1\%$ . Figures 6a and 6b also shows that while the walk-kick is executed, the robot still follows the support foot trajectory planned by the step planner, which in consequence, the walk trajectories are neither disrupted when the robot is walk-kicking nor does it become unstable. The area that is shaded in figures

6a and 6b is when the robot is walk-kicking. Figure 6a shows when the robot is doing a left kick and Figure 6b is the step trajectory when the robot is doing a right kick.

## 4 Discussion and Future Directions

The research conducted in this paper provided interesting results and possible research directions for the future. One result that needs further analysis is the strength of the back directions compared to the rest of the directions. It is possible that the momentum of different directions were variable, but the experiments were designed for the kick directions to have the same amount of momentum where the robot walks up to the ball and kicks with the same velocity in all directions. One possible explanation for a stronger back kick is the torso angle controller leaning forward when conducting a back kick to compensate for the instability causing the center of mass to shift towards the center generating greater angular momentum; this has to be verified with more experiments and analysis, i.e., more specifically analyzing the angular momentum output of the torso angle controller.

The momentum of the robot causes the stability of the robot to vary when it walk-kicks and an analysis of the instability in relation to the momentum of the robot needs to be considered to develop controllers that can deal with these types of forces. Currently in order to maintain stability, the robot has to decelerate because if the forward momentum of the robot increases, the robot may fall. Therefore, making the step position a parameter of the walk-kick controller while the robot kicks will definitely help with the stability. This entails optimizing the step planner to minimize instability while maintaining the reference trajectory.

The current controller for the walk-kick is open loop allowing the robot to passively use its momentum to generate kicks while walking, but the momentum of the robot and the kick trajectory acceleration can be actively controlled to generate a desired momentum allowing a closed-feedback loop system, i.e., this controller will also enable passing capabilities by controlling the force of the foot when it is in contact with the ball. Lastly, testing the dynamics of the walk-kick on different platforms will be important to determine the causes found in this paper and verify the system.

## 5 Conclusion

Although a variety of kick engines have been implemented over the years, there are none that demonstrate an omni-directional kick motion while walking. We therefore propose a walk-kick controller that enables a robot to kick in any direction while walking. The kick trajectories are generated by kick interpolators: cubic polynomials, sextic polynomials, and cubic Hermite splines. The kick trajectories are generated when the swinging foot is the kicking foot requested. The step planner handles the synchronization of both kick and step planners, while the swing controller communicates with the kick controller for the interpolation of the swinging foot. The balance

controller used for the walking engine uses the walking model as a black box, essentially enabling modularity in the controller to add features such as the kick controller, which grants the kicking trajectories to integrate with the walking engine without causing the robot to destabilize. The walk-kick controller was tested on a NAOv4 robot by kicking in ten directions (front, front-diagonal, side, back-diagonal, and back) for each foot. Our experiments verified that the walk-kick trajectories were consistent with an average absolute bearing of  $< 6^\circ$  which shows that the walk-kick is able to kick the ball in any direction while the robot is walking, and the CoM trajectories were also very stable within the bounds of the step trajectories. The step trajectories also followed the reference trajectories while executing the kick motions with a consistent average trajectory.

## References

1. Abdolmaleki, A., Simões, D., Lau, N., Reis, L.P., Neumann, G.: Learning a Humanoid Kick with Controlled Distance. In: Behnke, S., Sheh, R., Sarel, S., Lee, D.D. (eds.) *RoboCup 2016: Robot World Cup XX*. pp. 45–57. Springer International Publishing, Cham (2017)
2. Becht, I., de Jonge, M., Pronk, R.: A Dynamic Kick for the NAO Robot. Tech. rep., University of Amsterdam, Intelligent Robotics Lab (2013)
3. Böckmann, A., Laue, T.: Kick Motions for the NAO Robot Using Dynamic Movement Primitives. arXiv preprint arXiv:1606.00600 (2016)
4. Choi, J.Y., So, B.R., Yi, B.J., Kim, W., Suh, I.H.: Impact Based Trajectory Planning of a Soccer Ball in a Kicking Robot. In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. pp. 2834–2840 (April 2005). <https://doi.org/10.1109/ROBOT.2005.1570543>
5. Ferreira, R., Reis, L.P., Moreira, A.P., Lau, N.: Development of an Omnidirectional Kick for a NAO Humanoid Robot. In: Pavón, J., Duque-Méndez, N.D., Fuentes-Fernández, R. (eds.) *Advances in Artificial Intelligence – IBERAMIA 2012*. pp. 571–580. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
6. Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., Hirukawa, H.: Resolved Momentum Control: Humanoid Motion Planning based on the Linear and Angular Momentum. In: *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)* (Cat. No.03CH37453). vol. 2, pp. 1644–1650 vol.2 (Oct 2003). <https://doi.org/10.1109/IROS.2003.1248880>
7. Kim, D., Jorgensen, S.J., Stone, P., Sentis, L.: Dynamic Behaviors on the NAO Robot With Closed-Loop Whole Body Operational Space Control. In: *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. pp. 1121–1128 (Nov 2016). <https://doi.org/10.1109/HUMANOIDS.2016.7803411>
8. Lengagneua, S., Fraise, P., Ramdani, N.: Planning and Fast Re-Planning of Safe Motions for Humanoid Robots: Application to a Kicking Motion. In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 441–446 (Oct 2009). <https://doi.org/10.1109/IROS.2009.5354002>
9. Miossec, S., Yokoi, K., Kheddar, A.: Development of a software for motion optimization of robots - Application to the kick motion of the HRP-2 robot. In: *2006 IEEE International Conference on Robotics and Biomimetics*. pp. 299–304 (Dec 2006). <https://doi.org/10.1109/ROBIO.2006.340170>

10. Müller, J., Laue, T., Röfer, T.: Kicking a Ball – Modeling Complex Dynamic Motions for Humanoid Robots. In: del-Solar et al., R. (ed.) RoboCup 2010: Robot Soccer World Cup XIV, pp. 109–120. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
11. Peña, P.: An Omni-Directional Kick Engine for NAO Humanoid Robots. Master's thesis, University of Miami, 1365 Memorial Drive Coral Gables, FL. 33146 (January 2019)
12. Peña, P., Masterjohn, J., Visser, U.: An Omni-Directional Kick Engine for Humanoid Robots with Parameter Optimization. In: Akiyama, H., Obst, O., Sammut, C., Tonidandel, F. (eds.) RoboCup 2017: Robot World Cup XXI. Lecture Notes in Artificial Intelligence, vol. 11175, pp. 385–397. Springer International Publishing, Cham (2018). <https://doi.org/10.1007/978-3-030-00308-1>
13. Peña, P., Masterjohn, J., Visser, U.: Optimizing Kick Trajectory: A Comparative Study. In: Müller, C., Lisetti, C., Theobald, M. (eds.) GCAI 2017. 3rd Global Conference on Artificial Intelligence. EPiC Series in Computing, vol. 50, pp. 239–245. EasyChair (2017). <https://doi.org/10.29007/3f7v>, <https://easychair.org/publications/paper/Xw6m>
14. Seekircher, A., Visser, U.: An Adaptive LIPM-based Dynamic Walk using Model Parameter Optimization on Humanoid Robots. *KI* **30**(3), 233–244 (2016)
15. Sung, C.H., Kagawa, T., Uno, Y.: Planning of Kicking Motion with Via-Point Representation for Humanoid Robots. In: 2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI). pp. 337–342 (Nov 2011). <https://doi.org/10.1109/URAI.2011.6145987>
16. Wang, J., Liang, Z., Zhou, Z., Zhang, Y.: Kicking Motion Design of Humanoid Robots Using Gradual Accumulation Learning Method Based on Q-learning. In: 2016 Chinese Control and Decision Conference (CCDC). pp. 5274–5279 (May 2016). <https://doi.org/10.1109/CCDC.2016.7531941>
17. Wenk, F., Röfer, T.: Online Generated Kick Motions for the NAO Balanced Using Inverse Dynamics. In: Behnke, S., Veloso, M., Visser, A., Xiong, R. (eds.) RoboCup 2013: Robot World Cup XVII. pp. 25–36. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)
18. Xu, Y., Mellmann, H.: Adaptive Motion Control: Dynamic Kick for a Humanoid Robot. In: Dillmann, R., Beyerer, J., Hanebeck, U.D., Schultz, T. (eds.) KI 2010: Advances in Artificial Intelligence. pp. 392–399. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
19. Yi, S.J., McGill, S., Lee, D.D.: Online Kick Generation Method for Humanoid Soccer Robots. In: The 8th Workshop on Humanoid Soccer Robots (2013), [http://www.ais.uni-bonn.de/humanoidsoccer/ws13/papers/HSR13\\_Yi.pdf](http://www.ais.uni-bonn.de/humanoidsoccer/ws13/papers/HSR13_Yi.pdf)
20. Yi, S.J., McGill, S., He, Q., Hong, D., Lee, D.: Walk and Kick Motion Generation for a General Purpose Full Sized Humanoid Robot. In: Workshop on Humanoid Soccer Robots, IEEE-RAS International Conference on Humanoid Robots (Humanoids). IEEE (2014)