Progress report

O.A. Villarreal Magaña

February 20, 2017

1 Control structure for leg locomotion of HyQ

In this section the overview of the proposed strategy to control the locomotion pattern will be given. The overall control structured is depicted in Figure 1.

The scheme is comprised by a) a supervisory control that makes use of sensory information to provide a set of parameters (both in time and space) in order to achieve coordinated locomotion, b) a max-plus gait scheduler, that realizes the gait according to the time parameters assigned by the supervisory controller, c) a continuous reference generator that decides the shape of the trajectory of each of the legs in space, using the spacial parameters given by the supervisory controller. The continuous trajectory generator is preceded by an angular frequency continuous reference generator, which translates the time differences obtained via the time scheduler into a continuous angular frequency reference for each of the legs in order to achieve the proper synchronism for the desired gait. At the end of the structure a reference tracking controller is in charge of following the generated trajectory and imposes the input for each of the legs onto the robot.

There are three main feedback loops: one related to the reference tracking controller to correct the trajectory error, a second one related to the max-plus gait scheduler which detects the touchdown and lift-off events to update future timings in case of failure to reach or leave the ground, and the main feedback loop related to the supervisory controller which is in charge of modulating both the trajectory and time variables. In the following sections, each of the elements will be explained in more detail, except for the reference tracker and the robot plus environment blocks, which are not considered in this proposal.

1.1 Supervisory controller

The parameters given by the supervisory can be divided into two main groups: shape and time parameters. The shape parameters, determine the trajectory that each of the legs is going to follow, which in this case could be the step length L_s, step height H_s and the oscillator function that determines the "primitive" shape that of the trajectory. The time parameters on the other hand, determine the speed and more importantly, the timed events between each of the legs in order to achieve a coordinated
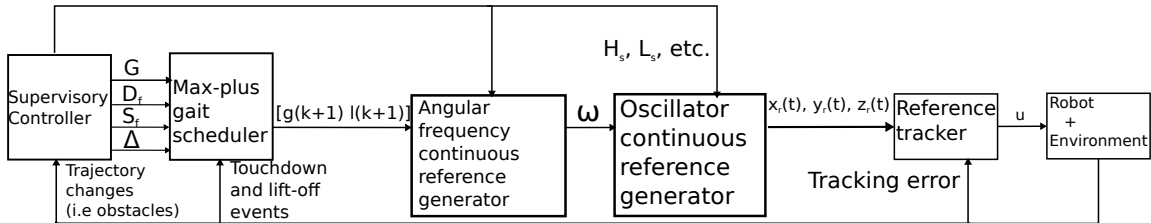

Figure 1: Control strategy proposed. The supervisory controller based on the information obtained from the robot and the environment, provides the parameters to the scheduler in order to generate a proper gait given the current conditions. Out of the max-plus scheduler, the lift-off and touchdown times are passed to the angular frequency generator, which defines the angular frequency that it is used for the oscillator-based reference generator. The references are followed by a tracking controller.

Table 1: Definition of leg indexes.

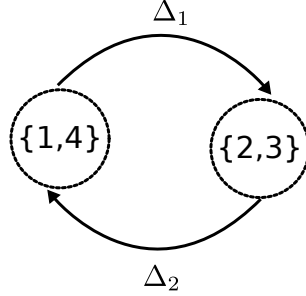| Leg	Leg index convention	Leg index number
Left front	LF	1
Right front	RF	2
Left hind	LH	3
Right hind	RH	4

Figure 2: Example of the switch between leg groups for a diagonal trotting gait. The time between the first group and the second group is Δ_1 and the time from the second group to the first one is Δ_2.

motion. These parameters are the duty factor D_f, the step frequency S_f, the gait pattern G and the synchronization time vector Δ.

The gait pattern $G = \{L_{1,1}, ..., L_{1,j}\} \prec ... \prec \{L_{r,1}, ..., L_{r,j}\}$ represents the order in which groups of legs L_j, for $j = 1, ..., n$ (n: number of legs), will move corresponding to an specific pattern. For example for a quadruped diagonal trotting gait, with the legs defined as in Table 1,one would write:

$$G = \{1, 4\} \prec \{2, 3\}, \tag{1}$$

likewise, a gait where only one leg at a time is in swing phase would be described as:

$$G = \{1\} \prec \{2\} \prec \{3\} \prec \{4\}. \tag{2}$$

The vector Δ is comprised by the time differences between each of the leg groups in G. These time differences are chosen by the supervisory control in order to achieve certain gait pattern. For example, for a trotting diagonal gait, Δ would be constructed by two elements:

$$\Delta = \begin{bmatrix} \Delta_1 & \Delta_2 \end{bmatrix}, \tag{3}$$

where Δ_1 and Δ_2 are the time differences between the first and the second leg group and the one corresponding from the second to the first, respectively, as depicted on Figure 2. These values represent the time each leg has to wait (or anticipate) to reach its lift-off event. If the leg needs to wait for an event, Δ_i would take a positive value, on the other hand if the leg needs to anticipate its lift-off time, Δ_i would assume a negative value.

The supervisory controller can use this set of time parameters to actively change the locomotion pattern depending on the situation that the robot is facing. Furthermore, these synchronization parameters in Δ can be designed according to a cost function related to an optimization problem, similar to what is done in [1] to choose the phase differences between legs. The advantage of this approach, is that this could potentially be modified online, since all the event dynamics are linear (this is, in the max-plus sense, and explained in the next section), would not require high amounts of computational time.

It has to be noted though, that the task of guaranteeing that "smooth" transitions between gaits is still open, since sudden and very diverse changes may cause odd and "limping" movements on the robot. One approach that could be considered is the one taken in [1, 2], where either a linear function was designed in order to achieve the transition in a smooth manner, or an optimization routine could be established as well in order to select a proper transition function between gait parameters from trot to gallop. In addition, to achieve a feasible gait pattern and to keep a desired duty factor D_f and step

frequency S_f, there are some constraints regarding the selection of Δ. For example, the total sum of each of the elements is subject to the following equation:

$$\sum_{i=1}^{n} \tau_i \leq \frac{1}{S_f}(1 - j(1 - D_f)) \text{ with n: number of leg groups.} \tag{4}$$

This expression constraints the total sum of time differences, since in order to keep a duty factor and a step frequency during the whole cycle, the sum of time differences cannot be greater than the step period $(\frac{1}{S_f})$ minus the time each legs spends in swing phase, multiplied by the number of legs j. There are also other limitations that need to be studied, since it has been noticed in simulation that there are also changes in the gait sequence related to the minimum value that each Δ_i can assume depending on the chosen duty factor. This is also a task to be tackled in the future.

1.2 Max-plus gait scheduler

Using the parameters given by the supervisory controller block, max-plus gait scheduler manages the time sequence for each of the legs in order to achieve the desired coordination. What comes out of the max-plus gait scheduler block are the touchdown and lift-off time events for each of the legs, which are obtained in a systematic manner making use only of the user input values given by the supervisory controller. Despite its simplicity, there are some advantages of using this type of systems to achieve coordination, since it represents a systematic method to assign the moments when each leg is going to move. Moreover, properties of transition and stability of the gait cycle can be analysed using max-plus linear systems theory.

Basics of max-plus algebra. As an initial step, the basic concepts of this so-called "tropical algebra" are explained based on [3]. Consider the following algebra:

$$(\Re_{max}, \oplus, \otimes, \varepsilon, e), \tag{5}$$

where:

$$\Re_{max} := \Re \cup \{-\infty\},$$
$$x \oplus y := max(x, y),$$
$$x \otimes y := x + y,$$
$$\varepsilon := -\infty,$$
$$e := 0.$$

In place, this algebra is defined by the set of real numbers including the negative infinity; the two binary operations: the maximum of two numbers and the addition; the absorbing element ε given by $-\infty$ and the identity element given by 0. Subsequently, max-plus algebra can be extended to matrices defined by the following structure:

$$(\Re_{max}^{n \times m}, \oplus, \otimes, \mathcal{E}, E), \tag{6}$$

with:

$$[A \oplus B]_{ij} = a_{ij} \oplus b_{ij} := \max(a_{ij}, b_{ij}),$$
$$[A \otimes C]_{ij} = \bigoplus_{k=1}^{m} a_{ik} \otimes c_{kj} := \max_{k=1,\ldots,m}(a_{ik} + c_{kj}),$$

where $A, B \in \Re_{max}^{n \times m}$, $C \in \Re_{max}^{m \times p}$, and the i, j element of A is denoted by $a_{ij} = [A]_{ij}$, and identity and zero matrices defined by:

$$[\mathcal{E}]_{ij} = \varepsilon,$$
$$[E]_{ij} = \begin{cases} e, & \text{if } i = j \\ \varepsilon, & \text{otherwise.} \end{cases}$$

3

Powers of matrices can also be defined as:

$$D^{\otimes k} := \underbrace{D \otimes D \otimes \ldots \otimes D}_{\text{k-times}}. \tag{7}$$

The max-plus algebra structure corresponds to a commutative idempotent semiring.

Max-plus algebra applied to legged locomotion. Max-plus linear systems have been used to model legged locomotion, abstracting the continuous-time motion of the legs into discrete event cycles [4]. Let $g_i(k)$ be the time instant when leg i touches the ground (touchdown time at time instant k) and $l_i(k)$ the time instant when it lifts-off from the ground (lift-off time at time instant k). We can describe the touchdown time as a function of the lift-off time plus the time instant that the leg stays in swing phase, i.e.:

$$g_i(k+1) = l_i(k+1) + \frac{1}{S_f}(1 - D_f). \tag{8}$$

Furthermore, the lift-off time instant can be defined as the previous touchdown time plus the time that the leg remains in stance phase,i.e.:

$$l_i(k+1) = g_i(k) + \frac{1}{S_f}D_f. \tag{9}$$

These two equations can be used iteratively to generate the movement of a single leg. In the case where several leg cycles want to be represented, we can define the following state vector:

$$X(k) = \begin{bmatrix} \underbrace{g_1(k) \quad \ldots \quad g_n(k)}_{g(k)} & \underbrace{l_1(k) \quad \ldots \quad l_n(k)}_{l(k)} \end{bmatrix}, \tag{10}$$

with n being the number of legs. By doing this, the leg cycles can be written in the following max-plus algebra linear system:

$$\begin{bmatrix} g(k+1) \\ l(k+1) \end{bmatrix} = \left[\begin{array}{c|c} \mathcal{E} & \frac{1}{S_f}(1-D_f) \otimes E \\ \hline \mathcal{E} & \mathcal{E} \end{array}\right] \otimes \begin{bmatrix} g(k+1) \\ l(k+1) \end{bmatrix} \oplus \left[\begin{array}{c|c} \mathcal{E} & \mathcal{E} \\ \hline \frac{1}{S_f}D_f \otimes E & \mathcal{E} \end{array}\right] \otimes \begin{bmatrix} g(k) \\ l(k) \end{bmatrix}. \tag{11}$$

where $\mathcal{E}$ and E are the identity and zero matrices as defined previously. If one wants to coordinate legs i and j, so that leg i lifts-off Δ_i seconds before (or after, if Δ_i is negative) after leg j has touched the ground. This can be expressed in the following relation:

$$l_i(k+1) = \max(g_i(k) + \frac{1}{S_f}D_f, g_j(k) + \Delta_i), \tag{12}$$

which can be written in max-plus algebra as:

$$l_i(k+1) = \begin{bmatrix} \frac{1}{S_f}D_f & \Delta_i \end{bmatrix} \otimes \begin{bmatrix} g_i(k) \\ g_j(k) \end{bmatrix}. \tag{13}$$

If coordination wants to be achieved for group of legs, a relation between the next lift-off time of a leg with the touchdown times of the other legs can be enforced, as done in the case of Equation (13). This can be written as a max-plus linear system using the state vector definition of Equation (10) in the following way:

$$\begin{bmatrix} g(k+1) \\ l(k+1) \end{bmatrix} = \left[\begin{array}{c|c} \mathcal{E} & \frac{1}{S_f}(1-D_f) \otimes E \\ \hline P & \mathcal{E} \end{array}\right] \otimes \begin{bmatrix} g(k+1) \\ l(k+1) \end{bmatrix} \oplus \left[\begin{array}{c|c} \mathcal{E} & \mathcal{E} \\ \hline \frac{1}{S_f}D_f \otimes E \oplus Q & \mathcal{E} \end{array}\right] \otimes \begin{bmatrix} g(k) \\ l(k) \end{bmatrix}, \tag{14}$$

where P and Q matrices represent matrices of time differences according to the gait pattern G and the synchronization vector Δ. This can be written in a more compact way as:

$$x(k+1) = R \otimes x(k+1) \oplus H \otimes x(k)$$

Up to this point it is not clear if Equation (14) can be solved explicitly (find $x(k+1)$ as a function dependent on $x(k)$). The solution to this problem is given in [4, ?] and provides the following equation:

$$x(k+1) = R^* \otimes H \otimes x(k),$$

and the existence of the matrix R^* is proven due to the fact that P is a nilpotent matrix. In fact, in [4], using the previously proposed parametrization of the gait space given by G, an explicit and systematic way to compute matrices Q and P was derived. A minor difference of the approach taken in this document is that, in the case of [4] all Δ_i values in the vector Δ were the same, whereas in here we try to give more versatility to possible number of gaits by giving different value to each Δ_i.

Property analysis of the gait cycle using max-plus algebra theory. As mentioned before, one of the advantages of using max-plus algebra systems is the possibility to study the properties of the cycle. If the matrix $A = R^*H$, one could find the max-plus eigenvalues and eigenvectors. It has been shown that matrix A has a unique eigenvalue, and it represents the total cycle time. On the other hand, the eigenvector is a representation of the steady state behaviour of the system. Another property investigated in previous studies is the so-called, coupling time refers to the number of finite steps that it takes for a system to reach a steady-state regime. This represents the transient behaviour, for example, while switching from one gait to another, it can be thought of in a similar way as the settling time for continuous-time systems. It has been proven that for a class of max-plus linear system the coupling time, denoted by k_0, is equal to 2. This means that it takes at most two cycle steps to reach max-plus steady-state when there is a perturbation.

This properties, have been used to provide ways for optimal gait switching, and to estimate the forward velocity of the robot in [4].

It has to be noted though, that some of these properties have been found to locomotion under very specific conditions (i.e. $\Delta_1 = ... = \Delta_i$, non-ballistic gaits with $\Delta_i > 0$, among others). Despite this fact, most of the research leaves as further work to investigate if this properties are valid for a wider range of locomotion conditions.

1.3 Angular frequency continuous reference generator

The angular frequency reference generator constitutes a key element in order to implement the timed events given by the max-plus scheduler into a continuous-time framework. Initially the trajectory is divided in two sections corresponding to the swing and stance phases, each with specific periods Ti_{st} and Ti_{sw}, respectively, and they are defined as follows:

$$Ti_{st} = l_i(k+1) - g_i(k) \tag{15}$$

$$Ti_{sw} = g_i(k+1) - l_i(k+1), \tag{16}$$

where $l_i(k)$ and $g_i(k)$ are the lift-off and touchdown times leg i at time event k. Figure 3 shows a drawing of the distinction between the two phases and the location where the touchdown and lift-off events take place. Using these two periods, we define an average angular frequency ω for the swing and stance phases. This angular frequency should be defined according to the trajectory that wants to be achieved, and it is defined as follows:

$$\bar{\omega} = \begin{cases} \frac{\pi}{Ti_{st}} \textbf{ for } t \in [g_i(k), l_i(k+1)] \\ \frac{\pi}{Ti_{sw}} \textbf{ for } t \in (l_i(k+1), g_i(k+1)] \end{cases} \tag{17}$$

The average angular frequency $\bar{\omega}$ is a parameter which ensures that the swing and stance phases will be reached at the appropriate time events in order to keep the synchronism of the gait. In this report, the
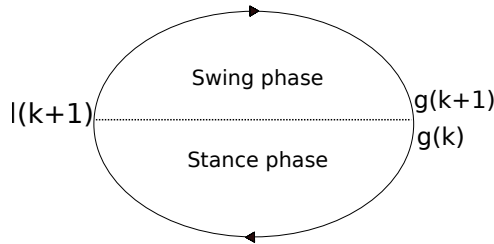

Figure 3: Phases of movement for an elliptical trajectory.

angular frequency is just defined as a constant value equal to the average angular frequency in 17. It would be desirable though, that this angular frequency was designed as a smooth transitioning function that would allow to keep the average along both swing and stance phase trajectories, thus complying with the following condition:

$$\frac{1}{Ti_p} \int^{Ti_p} \omega dt = \bar{\omega} \textbf{ for } p = st, sw. \tag{18}$$

Furthermore, it is also preferred that the movements during swing phase could be modulated, for example having faster motions in the beginning of the phase while slowing down towards the end. This situation could be approached in two different ways. The first one could be proposing another time event at the "highest" point of the ellipse, so one could have different functions for the angular frequency for the two halves of the swing phase. The second approach would be designing a profile function for ω in such a way that it complies with the condition given in 18. This second approach seems more appealing since it would imply not restating the max-plus algebra system (since the equations in Section 1.2 would no longer apply), and could potentially provide more freedom on how the movement in both phases is achieved, given a suitable design for the function of ω (i.e., via the solution of an optimization problem).

1.4 Oscillator continuous reference generator

The oscillator continuous reference generator corresponds to the equations given in [5] and a "squared" version of these equations used during simulation in this report. The main difference in this report lies on the fact that the coupling matrix used for synchrony is neglected here, since its function is being fulfilled by the max-plus algebra scheduler and the angular frequency continuous reference generator. The supervisory controller is in charge of defining the main shape parameters of the trajectory such as step length L_s and step height H_s. In this document we comprised these parameters in the variables a and b. The equations used for the elliptical case are given by:

$$x = \alpha(1 - \frac{x^2}{a^2} - \frac{z^2}{b^2})x + \frac{\omega a}{b}z \tag{19}$$

$$z = \beta(1 - \frac{x^2}{a^2} - \frac{z^2}{b^2})z - \frac{\omega b}{a}x, \tag{20}$$

and in the case of the "squared" oscillator are given by:

$$x = \alpha(1 - \frac{x^4}{a^4} - \frac{z^4}{b^4})x + \frac{1.18\omega a}{b^3}z^3 \tag{21}$$

$$z = \beta(1 - \frac{x^4}{a^4} - \frac{z^4}{b^4})z - \frac{1.18\omega b}{a^3}x^3. \tag{22}$$

1.5 Simulations

Using the previous scheme a simulation was performed consisting on the implementation of 2 different gaits, each of those with 2 variations where the gait parameters D_f and Δ are changed. These 4 variations, switch between each other at specific times throughout the process. The gait patterns are summarized in Table 2. The gait parameters are changed from Gait 1 to Gait 4 every 10 seconds. The purpose of this is first, prove that the proposed strategy is able to coordinate legs while providing a desired trajectory given by the oscillator equations, and second visualize the effects of changing gait parameters while the robot is moving.

Table 2: Gait parameters used in simulation

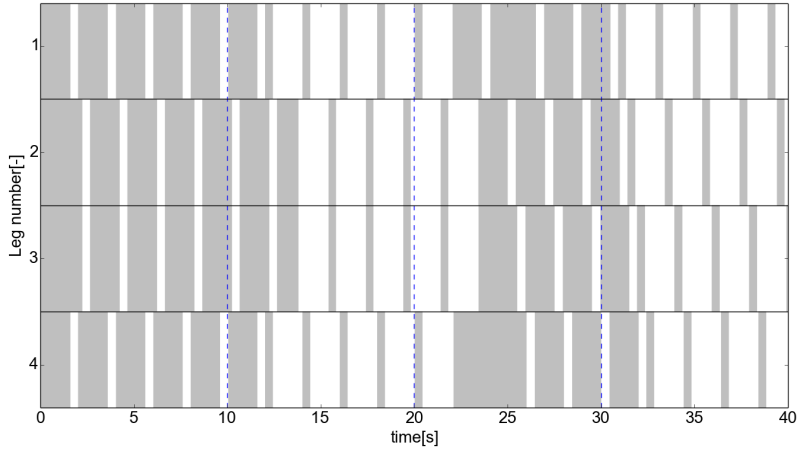|	G	D_f	$S_f[\frac{1}{s}]$	$\Delta[s]$			
Gait 1	$\{1,4\} \prec \{2,3\}$	0.8	0.5	$[0.24$	$0.48]$		
Gait 2	$\{1,4\} \prec \{2,3\}$	0.2	0.5	$[-0.1$	$-1.2]$		
Gait 3	$\{1\} \prec \{2\} \prec \{3\} \prec \{4\}$	0.8	0.5	$[0.1$	0.1	0.1	$0.1]$
Gait 4	$\{1\} \prec \{2\} \prec \{3\} \prec \{4\}$	0.2	0.5	$[-1.5$	-1.9	-1.1	$-1.1]$

Figure 4: Pattern plot corresponding to the simulation. Gray blocks represent that the leg is on the ground, while white space represents that the leg is not touching the ground. The blue dashed lines indicate the moments when a gait change is decided.

Figure 4 shows the pattern generated by the current strategy. In this plot it can be seen that regarding synchronization, the strategy is able to generate various types of gaits, of course it remains to be proven if these gaits are feasible, since up to this point we are not considering the dynamics of the robot itself. Furthermore, depending on how aggressive" the change of gait is and the difference between duty factors, the convergence to the new gait takes different times. This is a problem because if fast reactions are needed, there must be a reliable way to know how much will the gait switch will take. I personally believe that, if the change of paramaters is designed having a smooth transition in between gaits it is possibly to know the time that it will take to converge from one gait to the other, and furthermore, how far ahead one has to plan the changes on the gait parameters.

Considering the trajectory to be followed, in both cases (elliptical and squared) the proposed method was able to follow the reference if proper values of α and β are chosen (if these are not large enough the convergence properties of the oscillator may cause phase differences with respect to the desired trajectories). Figure 5 shows the evolution of reference coordinate z with respect to time from 5 to 25 seconds (the result is the same for the rest of the trajectory). It can be seen in the plot that the leg touches and leaves the ground at the specific times designated by the max-plus algebra scheduler (denoted by the dashed lines). Nevertheless, it still remains the task to deal first with disturbances such as tracking errors or the case were a touchdown or lift-off event is not reached (which is tackled by updating the max-plus algebra time scheduler).

Finally, a plot of the angular frequency ω with respect to time is shown in Figure 6. This plot shows that the angular frequency changes its constant value from swing to stance phase depending on the references given by the max-plus algebra time scheduler. The main issue with keeping constant the value of the angular frequency, is that at the beginning and ending of the "faster" phases(the swing and stance could last less or longer depending on the duty factor), the legs execute really aggressive movements. It is obvious that this could be improved if a function that respects the condition given in Equation 18 is designed to avoid this aggressive changes in angular frequency, providing more freedom to modulate de behaviour.
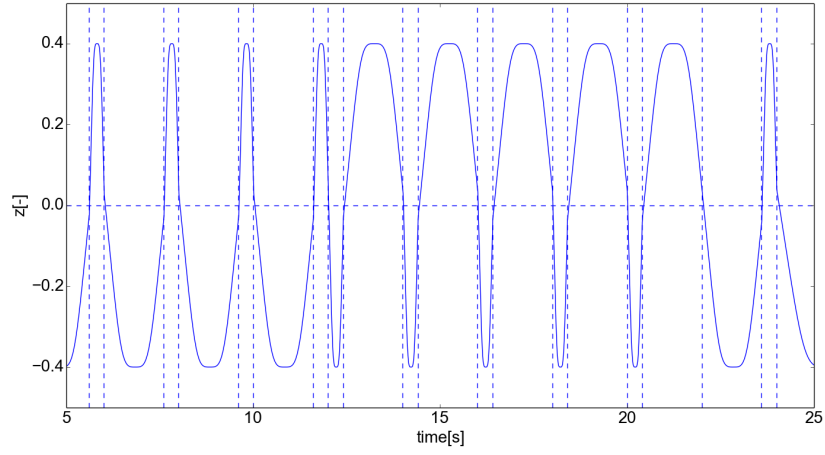
7

Figure 5: Plot showing the evolution of coordinate z with respect to time from 5 to 25 seconds. The dashed lines indicate the touchdown and lift-off times (i.e. the moments were the angular frequency is changed).
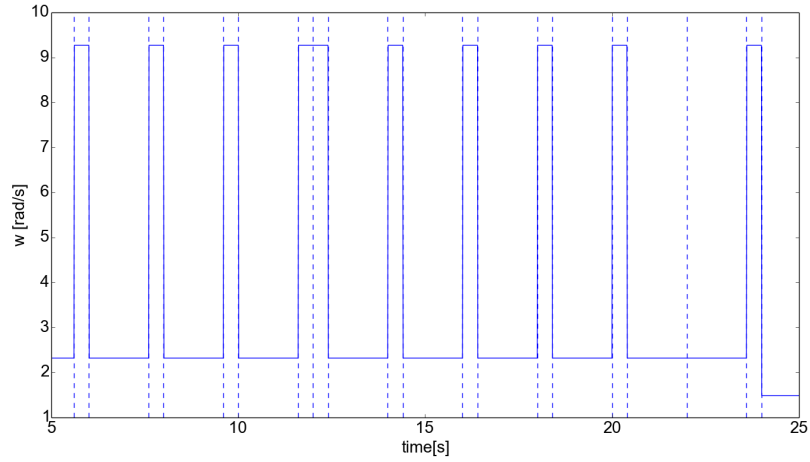

Figure 6: Angular frequency with respect to time. Dashed lines indicate touchdown and lift-off times.

References

[1] H.-W. Park and S. Kim, "Quadrupedal galloping control for a wide range of speed via vertical impulse scaling," *Bioinspiration & Biomimetics*, vol. 10, no. 2, p. 025003, 2015.

[2] U. Asif and J. Iqbal, "Motion Planning Using an Impact-Based Hybrid Control for Trajectory Generation in Adaptive Walking," *International Journal of Advanced Robotic Systems*, vol. 8, no. 4, pp. 212–224, 2011.

[3] B. D. Schutter and T. V. D. Boom, "Max-plus algebra and max-plus linear discrete event systems : An introduction," vol. 19, 2008.

[4] B. D. Schutter and A. J. J. V. D. Boom, "Switching max-plus models for legged locomotion," vol. 19, 2009.

[5] V. Barasuol, J. Buchli, C. Semini, M. Frigerio, E. R. De Pieri, and D. G. Caldwell, "A reactive controller framework for quadrupedal locomotion on challenging terrain," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2554–2561, 2013.