Delft University of Technology

Delft Center for Systems and Control

Technical report 09-045

Switching max-plus models for legged locomotion*

G.A.D. Lopes, R. Babuška, B. De Schutter, and A.J.J. van den Boom

If you want to cite this report, please use the following reference instead:
G.A.D. Lopes, R. Babuška, B. De Schutter, and A.J.J. van den Boom, "Switching max-plus models for legged locomotion," *Proceedings of the 2009 IEEE International Conference on Robotics and Biomimetics (ROBIO 2009)*, Guilin, China, pp. 221–226, Dec. 2009.

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: https://www.dcsc.tudelft.nl

Switching Max-Plus Models for Legged Locomotion

G.A.D. Lopes, R. Babuška, B. De Schutter and A.J.J. van den Boom

Abstract— **We present a new class of gait generation and control algorithms based on the Switching Max-Plus modeling framework that allows for the synchronization of multiple legs of walking robots. Transitions between stance and swing phases of each leg are modeled as discrete events on a system described by max-plus-linear state equations. Different gaits and gait parameters can be interleaved by using different system matrices. Switching in max-plus-linear systems offers a powerful collection of modeling, analysis, and control tools that, in particular, allow for safe transitions between different locomotion gaits that may involve breaking/enforcing synchronization or changing the order of leg lift off events. Experimental validation of the proposed algorithms is presented by the implementation of various horse gaits on a simple quadruped robot.**

I. INTRODUCTION

The field of discrete event systems (DES) [1] finds its core of applications in scheduling problems from engineering such as manufacturing, communications, traffic, computer systems, etc. Typically, the set of equations that describes the dynamics of such systems are nonlinear in traditional algebra. However, there is a subclass of timed DES (classes of discrete event systems where there exists an underlying time structure) that can be framed in sets of linear equations for a different type of algebra. These are called *max-plus-linear discrete event systems* (MPL-DES) defined in the *max-plus algebra* [2]–[4]. Systems that enforce synchronization and have no concurrency can be modeled in this framework. Systems that can be modeled as MPL-DES inherit a large set of analysis and control synthesis tools thanks to many parallels between the max-plus-linear systems theory and the traditional linear systems theory.

Legged locomotion systems can be elegantly modeled by limit cycles on cross products of circles, due to their intrinsic periodic nature (see "networks of phase oscillators" and "central pattern generators" (CPG) in Holmes et al. [5] or the earlier works of Grillner [6] and Cohen et al. [7]). One can represent the position of each leg at any given instant by mapping it into a phase in a circle. Synchronization can be achieved by enforcing phase differences on the circles. This modeling framework is very natural for both land and water locomotion (or even flight), but entails the construction of an "anchoring" map [8]. The biology community has extensively explored these concepts to classify the different gaits of horses [9], insects [10], and other animals [11]. The robotics community has harvested this knowledge to develop different techniques of generating motion in legged robots. Klavins et al. [12] show how to systematically generate vector fields that reach piecewise constant velocity limit cycles, Erden et al. [13] use reinforcement learning tools on a hexapod robot, and Zhao et al. [14] use CPG models for the control of a biomimetic fish. For an extensive literature review on legged locomotion see [5].

An interesting analogy can be made between enforcing phase differences in continuous cycles and synchronization in timed discrete event processes. In a typical (continuous) walking motion of a biped robot, the left leg should only lift off the ground after the right leg has touched down, to make sure the robot does not fall from lack of support[1]. This synchronization requirement can be modeled by the evolution of a discrete event system by abstracting each limit cycle in the circle into two sequential events in a closed circuit: lift off and touchdown. Each leg is then modeled by an event cycle and synchronization between legs is enforced by connections between each leg cycle. Synchronization of multiple legs is especially important for climbing robots [15], [16] where lack of support can result in catastrophic consequences. In this paper we show how the max-plus framework can be naturally utilized to systematically implement motion gaits for legged locomotion with guaranteed synchronization by design.

Some work has been done on low-level walking gait generation from a DES point of view such as [17], [18], and using Petri nets in [19]. In these implementations the focus is put on generating each individual gait. In this paper we take advantage of the properties of *switching max-plus-linear models* [20] to not only generate locomotion gaits but more importantly to deal with the transitions between different gaits and recovering from large perturbations. In a switching max-plus-linear system one can interchange different modes of operation. In each mode the discrete event system is described by a max-plus-linear state space model with different system matrices. In this application a mode corresponds to a specific gait.

Gait transition has been studied in biology from an energetic point of view [21] and in the robotics field was approached informally by Raibert et al. [22]. In this paper, we show that safe gait transitions arise naturally from the max-plus framework.

In Section II we introduce the tools behind max-plus systems, and we show in Section III how these can be used

Lopes, Babuška, De Schutter and van den Boom are with Delft Center for Systems and Control, Delft University of Technology, 2628 CD Delft, The Netherlands.

{g.a.delgadolopes, r.babuska, a.j.j.vandenboom} @tudelft.nl, b@deschutter.info

[1]In this paper we intentionally leave out locomotion gaits with aerial phases where robots/animals can spend most of their time with no support at all. We acknowledge their importance but defer their study for later publications.

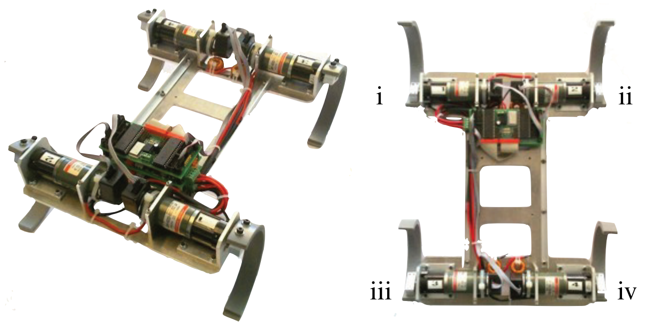

Fig. 1. Perspective and top view of a quadruped robot developed at DCSC, TU Delft, inspired by the hexapod robot RHex [23]. The roman numbers on the right picture represent the leg ordering.

to generate locomotion gaits for legged robots. In Section IV we review the equivalence between a class of timed Petri nets and the max-plus algebra and show how to construct locomotion gaits from timed Petri nets. We end by reporting on the experimental results obtained in a simple quadruped robot in Section VI.

II. Max-Plus algebra

We start by revising the structure of the max-plus algebra. Let $\varepsilon \equiv -\infty$, $e \equiv 0$, and $\mathbb{R}_{\max} = \mathbb{R} \cup \{\varepsilon\}$. Define the operations $\oplus, \otimes : \mathbb{R}_{\max} \times \mathbb{R}_{\max} \to \mathbb{R}_{\max}$ by:

$$x \oplus y \equiv \max(x, y)$$
$$x \otimes y \equiv x + y$$

Definition 1: The set $\mathbb{R}_{\max}$ with the operations $\oplus$ and $\otimes$ is called the max-plus algebra, denoted by $\mathcal{R}_{\max} = (\mathbb{R}_{\max}, \oplus, \otimes, \varepsilon, e)$.

Theorem 1 ([3]): The max-plus algebra $\mathcal{R}_{\max}$ has the algebraic structure of a commutative idempotent semiring.

The max-plus algebra can be interpreted as the traditional linear algebra with the operations '$+$' and '$\times$' replaced by the operators '$\max$' and '$+$', respectively, with the supplemental difference that the additive inverse does not exist, thus resulting in a semiring. Matrices can be defined by taking Cartesian products of $\mathbb{R}_{\max}$ and denoting $A \in \mathbb{R}_{\max}^{n\times m}$. Let $a_{ij} = [A]_{ij}$ be the i, j element of A. For $A, B \in \mathbb{R}_{\max}^{n\times m}$ and $C \in \mathbb{R}_{\max}^{m\times p}$ define the matrix sum $\oplus$ and matrix product $\otimes$ operations by:

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

Switching max-plus-linear systems are described by a state space model

$$x(k+1) = A_{(m(k))} \otimes x(k), \tag{1}$$

where the state $x(k)$ typically contains the time instants at which the internal events occur for the k-th time, and $A_{(m(k))}$ is the system matrix for mode $m(k)$. In this walking robot application the system matrix $A_{(m(k))}$ is in a set of locomotion gaits indexed by $m(k)$.

TABLE I

State variables and gait parameters.

$x(k)$	Full state vector of touchdown and lift off events.
$t_i(k)$	Touchdown time for leg i at iteration k.
$l_i(k)$	Lift off time for leg i at iteration k.
i	Index for legs.
θ_t	Leg touchdown angle.
θ_l	Leg lift off angle.
τ	Current time instant.
τ_f	Time leg spends in flight (swing).
τ_g	Time leg spends on the ground (stance).
τ_t, τ_p, τ_w	Time offset parameters for trotting, pacing and walking gaits respectively.
k_j	Index function for each state vector element.

III. Quadruped gaits

The first step towards modeling locomotion gaits is to define the state variables for the transition events. Let $l_i(k)$ be the time instant leg i lifts off the ground and $t_i(k)$ be the time instant it touches the ground, both for the k-th iteration. For a traditional alternating swing/stance gait one can impose that the time instant when the leg touches the ground must equal the time instant it lifted off the ground for the last time plus the time it stays in flight (denoted τ_f):

$$t_i(k) = l_i(k) + \tau_f \tag{2}$$

Analogously, we get a similar relation for the lift off time:

$$l_i(k) = t_i(k-1) + \tau_g, \tag{3}$$

where τ_g is the stance time and t_i uses the previous iteration such that equations (2) and (3) can be used iteratively. Suppose now that one aims to synchronize leg i with leg j in such a way that leg i can only lift off τ_Δ seconds after leg j has touched the ground. One can then write the relation:

$$\begin{aligned} l_i(k) &= \max\left(t_i(k-1) + \tau_g, t_j(k-1) + \tau_\Delta\right) \\ &= \begin{bmatrix} \tau_g & \tau_\Delta \end{bmatrix} \otimes \begin{bmatrix} t_i(k-1) \\ t_j(k-1) \end{bmatrix}. \end{aligned} \tag{4}$$

Equation (4) enforces simultaneously that both the leg i stays at least τ_g seconds in stance and will only lift off at least τ_Δ seconds after leg j has touched down. When both conditions are satisfied, lift off takes place. Following this reasoning, one can efficiently represent motion gaits in terms of synchronization of timed events. Figure 2 illustrates three different gaits for quadrupeds: pacing, trotting, and walking (according to the leg numbering in Figure 1), and Table I compiles the state variables and the associated parameters used throughout this paper. Pacing, commonly used by camels and dromedaries, is a lateral two-beat gait, i.e., front and back left legs are synchronized and opposite in phase to front and back right legs. In trotting, opposite legs are synchronized and we assume that for the walking gait at least three legs are on the ground at all times.

The traditional pacing gait with no aerial phase is illustrated in Figure 2.a). The arrows represent the relationship between events that must occur for leg lift off to happen.

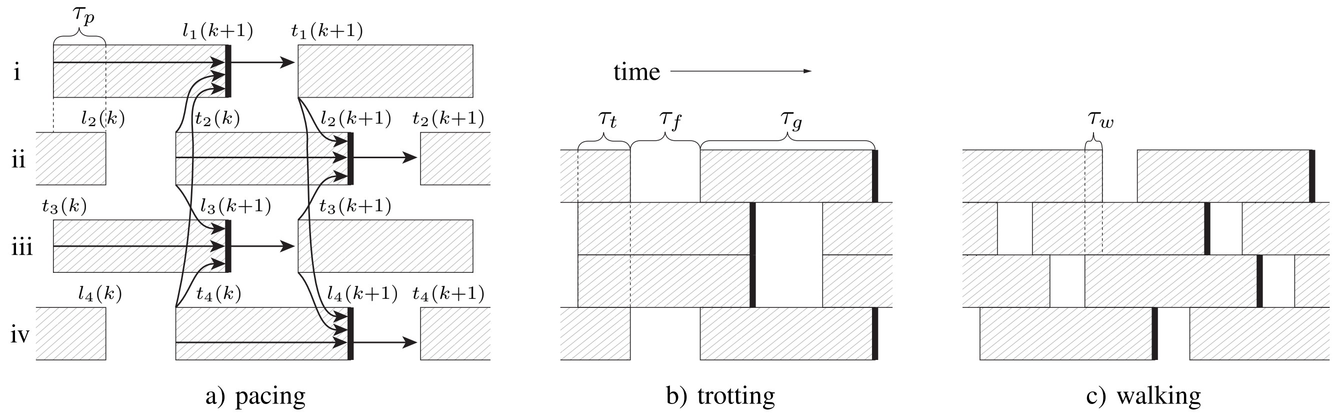

Fig. 2. Time evolution of gaits for a quadruped. The hatched boxes represent the leg stance, and the solid thick lines represent the lift off events $l_i(k+1)$. The arrows in a) represent events that must occur for each lift off and touchdown events to happen.

Following the notation in Figure 2.a) one obtains the timed event equations for the pacing gait:

$$\begin{aligned}
t_1(k+1) &= l_1(k+1) + \tau_f \\
l_1(k+1) &= \max\left(t_1(k)+\tau_g, t_2(k)+\tau_p, t_4(k)+\tau_p\right) \\
t_2(k+1) &= l_2(k+1) + \tau_f \\
l_2(k+1) &= \max\left(t_2(k)+\tau_g, t_1(k+1)+\tau_p, t_3(k+1)+\tau_p\right) \\
t_3(k+1) &= l_3(k+1) + \tau_f \\
l_3(k+1) &= \max\left(t_3(k)+\tau_g, t_2(k)+\tau_p, t_4(k)+\tau_p\right) \\
t_4(k+1) &= l_4(k+1) + \tau_f \\
l_4(k+1) &= \max\left(t_4(k)+\tau_g, t_1(k+1)+\tau_p, t_3(k+1)+\tau_p\right).
\end{aligned}$$

The previous set of equations can be transformed into the structure of equation (1) by recursive substitution. For example, the update equation for t_1 becomes:

$$\begin{aligned}
t_1(k+1) &= l_1(k+1) + \tau_f \\
&= \max\left(t_1(k)+\tau_g, t_2(k)+\tau_p, t_4(k)+\tau_p\right) + \tau_f \\
&= \max\left(t_1(k)+\tau_g+\tau_f, t_2(k)+\tau_p+\tau_f, t_4(k)+\tau_p+\tau_f\right)
\end{aligned}$$

Following the same recursive process for the other variables, and defining the state variables $x(k) \in \mathbb{R}^8_{\max}$ by

$$x(k) = [t_1(k)\ l_1(k)\ \cdots\ t_4(k)\ l_4(k)]^T,$$

one can find the max-plus-linear system matrix for the pacing gait, that we denote A_p. For gait symmetry we assume that

$$\tau_g > \tau_f, \text{ and } \tau_p = \frac{\tau_g - \tau_f}{2}.$$

The extra parameters $\tau_{fg} = \tau_f + \tau_g$ and $\tau_{pf} = \tau_p + \tau_f$ are introduced for compactness.

$$A_p = \begin{bmatrix}
\tau_{fg} & \varepsilon & \tau_{pf} & \varepsilon & \varepsilon & \varepsilon & \tau_{pf} & \varepsilon \\
\tau_g & \varepsilon & \tau_p & \varepsilon & \varepsilon & \varepsilon & \tau_p & \varepsilon \\
\tau_{pf}+\tau_{fg} & \varepsilon & \tau_{fg} & \varepsilon & \tau_{pf}+\tau_{fg} & \varepsilon & \tau_{fg} & \varepsilon \\
\tau_{pf}+\tau_g & \varepsilon & \tau_g & \varepsilon & \tau_{pf}+\tau_g & \varepsilon & \tau_g & \varepsilon \\
\varepsilon & \varepsilon & \tau_{pf} & \varepsilon & \tau_{fg} & \varepsilon & \tau_{pf} & \varepsilon \\
\varepsilon & \varepsilon & \tau_p & \varepsilon & \tau_g & \varepsilon & \tau_p & \varepsilon \\
\tau_{pf}+\tau_{fg} & \varepsilon & \tau_{fg} & \varepsilon & \tau_{pf}+\tau_{fg} & \varepsilon & \tau_{fg} & \varepsilon \\
\tau_{pf}+\tau_g & \varepsilon & \tau_g & \varepsilon & \tau_{pf}+\tau_g & \varepsilon & \tau_g & \varepsilon
\end{bmatrix}$$

Solving system (1) for the pacing gait A_p with all time events initialized to zero, and with parameters $\tau_g = 3$s and $\tau_f = 1$s, results in the following leg scheduling:

k	t_1	l_1	t_2	l_2	t_3	l_3	t_4	l_4
1	0	0	0	0	0	0	0	0
2	4	3	6	5	4	3	6	5
3	8	7	10	9	8	7	10	9
4	12	11	14	13	12	11	14	13
$\vdots$	$\vdots$							$\vdots$

After a single iteration all the synchronization conditions are met. See Appendix I for the system matrices of trotting and walking gaits.

IV. Timed event graphs

The theory of Petri nets [1] provides is an intuitive tool for representing DES graphically: the *Petri net graphs*. It is straightforward to generate walking gaits by evolving a Petri net in time if: 1) each leg is modeled as a circuit with lift off and touchdown transitions 2) synchronizations between legs are implemented by firing conditions on the lift off transition. Moreover, by carefully designing the Petri net one can take advantage of the analysis and synthesis tools of max-plus algebras. A subclass of timed Petri nets called *event graphs* are formally equivalent to max-plus linear systems [3]. This allows for the intuitive design of locomotion gaits by event graphs with a direct translation to max-plus linear systems and vice-versa.

Definition 2 ([4]): A timed Petri net $\mathcal{G}$ is characterized by a set of places $\mathcal{P}$, a set of transitions $\mathcal{Q}$, a set of arcs $\mathcal{D}$ from transitions to places and vice versa, an initial marking $\mathcal{M}_0$, and a holding time vector $\mathcal{T}$. If each place has exactly one upstream and one downstream transition, then the timed Petri net is called a *timed event graph*.

Figure 3 illustrates the timed event graph for the pacing gait described in Figure 2. The construction of such a graph goes as follows:

1) For each leg define a circuit with the two events: touchdown T_i and lift off L_i. Between the events add the

places G_i for the time the leg i stays in the ground and F_i for the time it is in flight. In Figure 3 this is represented by the four numbered thicker circuits.

2) For each required synchronization add a place between transitions. It is important to make sure that each place has a unique upstream arc and a unique downstream arc. For example, the place labeled S_{14} enforces that the lift off transition of leg 'iv' only fires after leg 'i' has touched the ground.

3) Initialize the marking such that all the ground places have a token (the robot starts by having all the legs on the ground). Moreover, tokens are added to synchronization places such that all (closed) circuits are alive and the initial marking is feasible. For the case of the pacing gait, four tokens are added such that the lift off transitions of legs 'ii' and 'iv' are ready to fire in the initial marking.

Following the steps presented above, it is possible to reproduce many locomotion gaits. Once the timed event graph is constructed it is straightforward to find its associated max-plus linear system [4]:

1) Each transition Ψ_i in the timed event graph is assigned to a state variable ψ_i in the max-plus algebra.
2) For each incoming arc to a transition Ψ_i make a list of

- the event Ψ_j that precedes the event Ψ_i for that arc and add j to a set $\mathcal{S}$
- the time constant υ_j of the place of the arc's origin
- the number of tokens κ_j of the place of the arc's origin

Then write the expressions:

$$\psi_i(k+1) = \max_{j \in \mathcal{S}} \left(\psi_j(k+1-\kappa_j) + \upsilon_j \right) \tag{5}$$

For example, consider the transition T_1 in Figure 3. It has a single incoming arc from the transition L_1 with time constant τ_f and zero tokens. Thus, equation (5) for this transition is

$$t_1(k+1) = l_1(k+1) + \tau_f$$

For transition L_1 we have three incoming arcs from T_1, T_2, and T_4, with respective time constants (τ_g, τ_p, τ_p) and one token per place. The transition equation for l_2 is then:

$$l_1(k{+}1) = \max\left(t_1(k){+}\tau_g, t_2(k){+}\tau_p, t_4(k){+}\tau_p\right)$$

Following this procedure one obtains exactly the transition equations for the pacing gait derived in Section III. Note that if the four synchronization tokens added in the initial markup to the places $\{S_{21},S_{23},S_{43},S_{41}\}$ were instead placed in $\{S_{12},S_{14},S_{32},S_{34}\}$ the resulting equations would be a change of coordinates away from the pacing equations in Section III. The same would be true for any resulting token configuration after any feasible firing of the timed event graph in Figure 3. The remaining combinations of tokens either block the Petri net or break the required synchronization resulting in a incorrect time evolution for the gait.

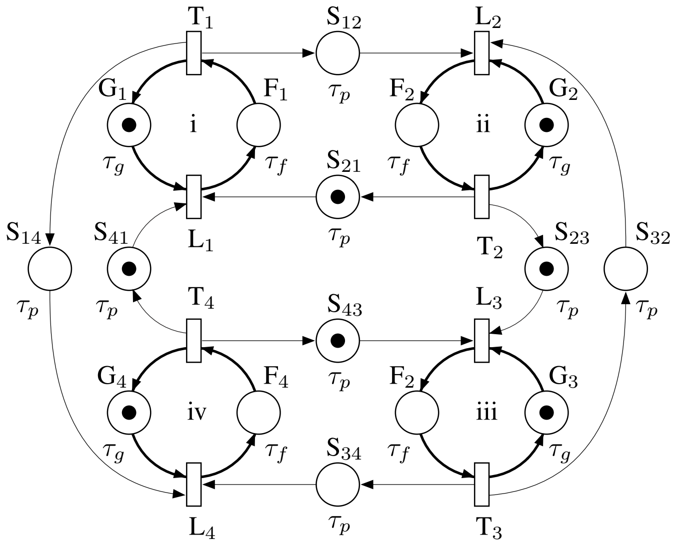

Fig. 3. Timed event graph for a pacing gait. Leg lift off events are represented by L_i and touchdown events by T_i. Each place has a label name on top and a holding time on the bottom. Ground places are represented by G_i, flight places by F_i, and leg synchronization places by S_i. The thick numbered loops represent the discrete event periodic cycles for each leg.

V. Control Structure

The max-plus-linear system for locomotion gaits derived in Section III returns a state vector of the time instants when leg touchdown and lift off events must occur in the future. These must be translated into continuous-time trajectories for the control of each motor. Moreover, the discrete event state variables must be updated by the continuous-time state variables (angles of the legs) to measure the true time the events occurred and recompute the future event timings. Figure 4 presents the block diagram of the hybrid control structure we propose in this paper. The *supervisory control*[2] block generates max-plus-linear system matrices. The *max-plus gait scheduler* block implements system (1). The *continuous time scheduler* generates a continuous time reference trajectory for the events generated by the max-plus scheduler. Finally, the *reference trajectory tracker* implements local PD controllers at each motor to track the reference trajectory.

A. Continuous-time trajectory generator

The timed event equations derived in Section III must be mapped into the continuous-time domain of a legged robot. This can be accomplished by defining a reference trajectory generating function

$$\theta_{\mathrm{ref}} : \mathbb{R}^+ \times (\mathbb{R}^{2n}_{\max})^p \to (S^1)^n$$

that takes as inputs time $\tau \in \mathbb{R}^+$ plus a collection of p discrete events $x(k) \in \mathbb{R}^{2n}_{\max}$ and outputs a piecewise linear

[2] In this paper the supervisory control box is replaced by a human selecting different locomotion gaits through a graphical user interface.

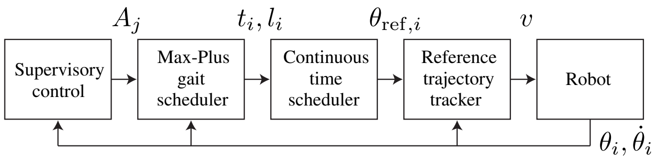

Fig. 4. Block diagram of control structure for a legged robot.

trajectory for each of the leg angles, with $\theta_t < \theta_l$:

$$\theta_{\mathrm{ref},i}(\tau) := \begin{cases} \dfrac{\theta_l\,(t_i(k_{2i-1}) - \tau) + (\theta_t + 2\pi)\,(\tau - l_i(k_{2i}))}{t_i(k_{2i-1}) - l_i(k_{2i})} \\ \qquad\qquad \text{if } \tau \in [l_i(k_{2i}), t_i(k_{2i-1})) \\ \dfrac{\theta_t\,(l_i(k_{2i}+1) - \tau) + \theta_l\,(\tau - t_i(k_{2i-1}))}{l_i(k_{2i}+1) - t_i(k_{2i-1})} \\ \qquad\qquad \text{if } \tau \in [t_i(k_{2i-1}), l_i(k_{2i}+1)) \end{cases}$$

The function θ_{ref} takes a p-collection of events since there is no necessity for the intervals $[l_i(k_{2i}), t_i(k_{2i-1}))$ and $[t_i(k_{2i-1}), l_i(k_{2i}+1))$ to be overlapping for all legs. The event indices $\{k_j\} \in \mathbb{N}^8$ are chosen for each leg such that the time τ lies in the proper interval.

B. Feedback control

Feedback control is implemented in both the *reference trajectory tracker* and *max-plus gait scheduler* blocks represented in the diagram of Figure 4. In the first, a reference trajectory feedback control loop is implemented by a simple PD controller. In the second, events of touchdown and lift off are measured by observing when the legs cross specific angles during their motion. The time the true event occurred is updated into the max-plus gait scheduler which then recomputes the subsequent event transition times. This way, e.g., if one leg is stopped while in the air, the others will not lift off, guaranteeing that the robot does not lose support.

C. Switching gaits

As previously described, each gait is encoded in the system matrices $A_{(m(k))}$ of equation (1). In this paper we have

$$A_{(m(k))} \in \mathcal{A} = \{A_p, A_t, A_w\},$$

Where the indices represent pacing, trotting and walking. The max-plus algebraic representation guarantees that each transition is safe by construction. Note that the system matrices are parameterized by the time constants that encode leg flight time, ground time, and other leg offsets:

$$A_{(m(k))} = A_{(m(k))}(\tau_f, \tau_g, \dots).$$

Thus, beyond changing gaits, one can also safely change the parameters of A at each iteration k, resulting in different locomotion speeds or different leg offset synchronizations. By changing at each iteration k the lift off and touchdown angles θ_l, θ_t in the parameterization of the reference trajectory generating function $\theta_{\mathrm{ref},i}(\tau)$, one can adapt the gait for difference slopes, or steer the robot by introducing different angle offsets from the left to the right legs.

VI. Experimental results

The tools presented in this paper were implemented in a small quadruped robot developed at TU Delft, illustrated in Figure 1. The morphology is inspired by the hexapod RHex robot [23] albeit with a different number of legs. The algorithms are implemented in Matlab, running on a small form factor laptop PC that stands on the robot. A Matlab GUI is used to switch the locomotion gaits and their associated parameters. Figure 5 illustrates a sample forward locomotion experiment where different gaits are sequenced. In this experiment legs 'i' and 'iii' are held by hand in the air to illustrate the robustness of the implementation. Leg 'i' is held in the air during a gait switch. As seen in the iteration 6 of Figure 5, gait transitions are not guaranteed to be time optimal, but they are safe.

VII. Conclusions and Future Work

We have presented a new modeling tool for locomotion of multi-legged robots based on the max-plus algebra. Gait design and implementation can be efficiently accomplished by connected circuits of discrete events. Leg synchronization is guaranteed by design of the max-plus framework. We show that gait switching is efficiently implemented by simply switching system matrices in the evolution of a max-plus-linear system. Experimental results in locomotion are presented for validation of the proposed framework.

Although the present implementation uses the max-plus framework as a modeling technique that achieves a compact notation and efficient implementation, we aim in future work to fully explore the analysis and synthesis tools of the max-plus algebras to improve on the switching behaviors of legged locomotion beyond safety. As an example, by looking at the eigenstructure of the system matrices of a max-plus-linear system it is possible extract very useful data that can inform a supervisory control of the "best"[3] gaits to switch to, at any given instant. We are currently developing such a supervisor controller.

VIII. Acknowledgments

We thank Sait Izmit at Philips Applied Technologies for providing and supporting the real-time 'RT-Motion USB' motor control boards and the low-level software framework for the quadruped robot.

Appendix I
System matrices for trotting and walking gaits

A. Trotting gait

The trotting gait, illustrated in Figure 2.b) is analogous to the pacing gait. Again we choose $\tau_t = (\tau_g - \tau_f)/2$ and

[3] In terms of switching speed, i.e. pick the gait that introduces the least amount of delay into the event transitions.

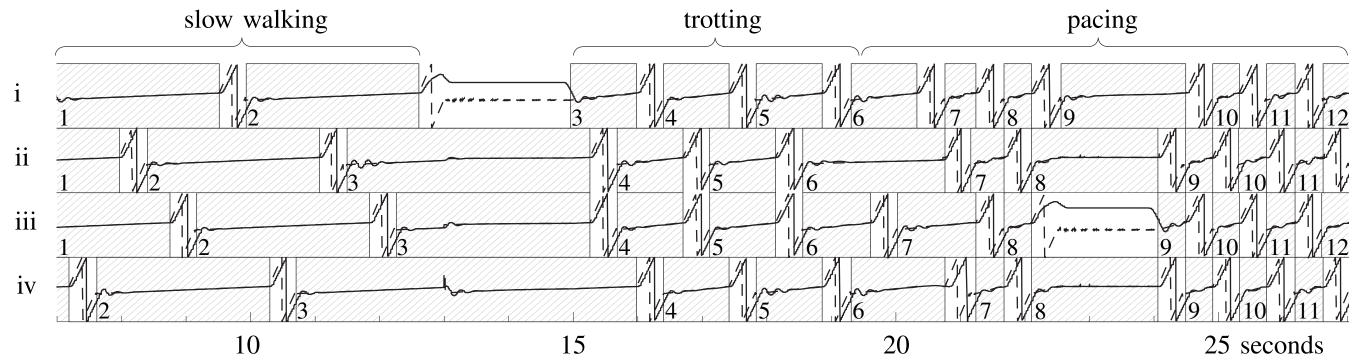

Fig. 5. Experiment of time evolution of three sequential locomotion gaits with safe transitions. The hatched boxes represent the stance time. The numbers inside the hatched boxes represent the k-th iteration of system (1). Dashed lines represent the reference trajectory and solid lines represent the real leg positions. The height of the box represents the range $[-\pi, \pi]$ with the origin at the bottom position of the leg. During each flight phase the angle wraps around, resulting in the vertical lines in the trajectories. Leg 'i' is held by hand in the air between iterations 2 and 3 and leg 'iii' is held between iterations 8 and 9.

$\tau_{tf} = \tau_t + \tau_f$. The system matrix A_t for this gait is:

$$A_t = \begin{bmatrix} \tau_{fg} & \varepsilon & \tau_{tf}+\tau_{fg} & \varepsilon & \tau_{tf}+\tau_{fg} & \varepsilon & \tau_{fg} & \varepsilon \\ \tau_g & \varepsilon & \tau_{tf}+\tau_g & \varepsilon & \tau_{tf}+\tau_g & \varepsilon & \tau_g & \varepsilon \\ \tau_{tf} & \varepsilon & \tau_{fg} & \varepsilon & \varepsilon & \varepsilon & \tau_{tf} & \varepsilon \\ \tau_t & \varepsilon & \tau_g & \varepsilon & \varepsilon & \varepsilon & \tau_t & \varepsilon \\ \tau_{tf} & \varepsilon & \varepsilon & \varepsilon & \tau_{fg} & \varepsilon & \tau_{tf} & \varepsilon \\ \tau_t & \varepsilon & \varepsilon & \varepsilon & \tau_g & \varepsilon & \tau_t & \varepsilon \\ \tau_{fg} & \varepsilon & \tau_{tf}+\tau_{fg} & \varepsilon & \tau_{tf}+\tau_{fg} & \varepsilon & \tau_{fg} & \varepsilon \\ \tau_g & \varepsilon & \tau_{tf}+\tau_g & \varepsilon & \tau_{tf}+\tau_g & \varepsilon & \tau_g & \varepsilon \end{bmatrix}$$

B. Walking gait

For the walking gait, illustrated in Figure 2.c) we assume that at least three legs must be on the ground at all times. Choosing the timing parameter

$$\tau_g > 3\tau_f, \qquad \tau_w = \frac{\tau_g - 3\tau_f}{4}$$

results in a symmetrical gait. Defining $\tau_{wf} = \tau_w + \tau_f$, the system matrix A_w for the walking gait is:

$$\begin{bmatrix} 4\tau_{wf} & \varepsilon & 6\tau_{wf} & \varepsilon & 5\tau_{wf} & \varepsilon & 3\tau_{wf}+\tau_{fg} & \varepsilon \\ 3\tau_{wf}+\tau_w & \varepsilon & 5\tau_{wf}+\tau_w & \varepsilon & 4\tau_{wf}+\tau_w & \varepsilon & 3\tau_{wf}+\tau_g & \varepsilon \\ 2\tau_{wf} & \varepsilon & 4\tau_{wf} & \varepsilon & 3\tau_{wf} & \varepsilon & \tau_{wf}+\tau_{fg} & \varepsilon \\ \tau_{wf}+\tau_w & \varepsilon & 3\tau_{wf}+\tau_w & \varepsilon & 2\tau_{wf}+\tau_w & \varepsilon & \tau_{wf}+\tau_g & \varepsilon \\ 3\tau_{wf} & \varepsilon & 5\tau_{wf} & \varepsilon & 4\tau_{wf} & \varepsilon & 2\tau_{wf}+\tau_{fg} & \varepsilon \\ 2\tau_{wf}+\tau_w & \varepsilon & 4\tau_{wf}+\tau_w & \varepsilon & 3\tau_{wf}+\tau_w & \varepsilon & 2\tau_{wf}+\tau_g & \varepsilon \\ \tau_{wf} & \varepsilon & 3\tau_{wf} & \varepsilon & 2\tau_{wf} & \varepsilon & \tau_{fg} & \varepsilon \\ \tau_w & \varepsilon & 2\tau_{wf}+\tau_w & \varepsilon & \tau_{wf}+\tau_w & \varepsilon & \tau_g & \varepsilon \end{bmatrix}$$

References

[1] C. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Kluwer, 1999.

[2] R. Cuninghame-Green, *Minimax Algebra*, ser. Lecture Notes in Economics and Mathematical Systems. Springer-Verlag, 1979, vol. 166.

[3] F. Baccelli, G. Cohen, G. Olsder, and J. Quadrat, *Synchronization and Linearity: An Algebra for Discrete Event Systems*. Wiley, 1992.

[4] B. Heidergott, G. Olsder, and J. van der Woude, *Max Plus at Work: Modeling and Analysis of Synchronized Systems*. Kluwer, 2006.

[5] P. Holmes, R. Full, D. Koditschek, and J. Guckenheimer, "The dynamics of legged locomotion: Models, analyses, and challenges," *SIAM Review*, vol. 48, no. 2, pp. 207–304, 2006.

[6] S. Grillner, "Neurobiological bases of rythmic motor acts in vertebrates," *Science*, vol. 228, pp. 143–149, 1985.

[7] A. Cohen, S. Rossignol, and S. Grillner, *Neural Control of Rhyhtmic Movements in Vertebrates*. Wiley, 1988.

[8] R. Full and D. Koditschek, "Templates and anchors: Neuromechanical hypotheses of legged locomotion on land," *J. of Experimental Biology*, vol. 202, no. 23, pp. 3325–3332, 1999.

[9] M. Hildebrand, "Symmetrical gaits of horses," *Science*, vol. 150, no. 3697, pp. 701–708, 1965.

[10] D. M. Wilson, "Insect walking," *Annual Review of Entomology*, vol. 11, pp. 103–122, 1966.

[11] R. Alexander, "The Gaits of Bipedal and Quadrupedal Animals," *Int. J. of Robotics Research*, vol. 3, no. 2, pp. 49–59, 1984.

[12] E. Klavins and D. Koditschek, "Phase regulation of decentralized cyclic robotic systems," *Int. J. of Robotics Research*, vol. 21, no. 3, pp. 257–275, 2002.

[13] M. S. Erden and K. Leblebicioğlu, "Free gait generation with reinforcement learning for a six-legged robot," *Robotics and Autonomous Systems*, vol. 56, pp. 199–212, 2008.

[14] W. Zhao, Y. Hu, L. Zhang, and L. Wang, "Design and cpg-based control of biomimetic robotic fish," *Control Theory and Applications, IET*, vol. 3, no. 3, pp. 281–293, 2009.

[15] L. Guo, K. Rogers, and R. Kirkham, "A climbing robot with continuous motion," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 1994, pp. 2495–2500.

[16] K. Autumn, M. Buehler, M. Cutkosky, R. Fearing, R. J. Full, D. Goldman, R. Groff, W. Provancher, A. A. Rizzi, U. Saranli, A. Saunders, and D. E. Koditschek, "Robotics in scansorial environments," in *Unmanned Ground Vehicle Technology VII*, G. R. Gerhart, C. M. Shoemaker, and D. W. Gage, Eds., vol. 5804, no. 1. SPIE, 2005, pp. 291–302.

[17] M. Antoniotti and B. Mishra, "Discrete event models+temporal logic=supervisory controller: automatic synthesis of locomotion controllers," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, vol. 2, 1995, pp. 1441–1446.

[18] N. Suzuki, K. Takahira, and H. Kajiwara, "Development of a des toolbox and its application to a robot-gait planning," in *Proc. of Soc. of Instrument and Control Eng.*, vol. 2, 2002, pp. 833–834.

[19] Z. Guangtao, Z. Haojun, W. Jinsong, and L. Tiemin, "Petri-net-based coordination motion control for legged robot," in *Proc. of IEEE Int. Conf. on Systems, Man and Cybernetics*, vol. 1, 2003, pp. 581–586.

[20] T. van den Boom and B. De Schutter, "Modelling and control of discrete event systems using switching max-plus-linear systems," *Control Engineering Practice*, vol. 14, no. 10, pp. 1199–1211, 2006.

[21] D. Hoyt and C. Taylor, "Gait and the energetics of locomotion in horses," *Nature*, vol. 292, pp. 239–240, 1981.

[22] M. Raibert, H. B. Jr., M. Chepponis, J. Koechling, J. Hodgins, D. Dustman, W. Brennan, D. Barrett, C. Thompson, J. Hebert, W. Lee, and L. Borvansky, "Dynamically stable legged locomotion," MIT, Tech. Rep. 1179, 1989.

[23] U. Saranli, M. Buehler, and D. E. Koditschek, "Rhex: A simple and highly mobile hexapod robot," *Int. J. of Robotics Research*, vol. 20, no. 7, pp. 616–631, 2001.