Technical report 11-034

Optimal gait switching for legged locomotion

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

If you want to cite this report, please use the following reference instead:

URL: https://www.dcsc.tudelft.nl

*This report can also be downloaded via https://pub.deschutter.info/abs/11_034.html
Optimal gait switching for legged locomotion

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

Abstract—Switching gaits in many-legged robots can present challenges due to the combinatorial nature of the gait space. In this paper we present an intrinsically safe gait switching generator that minimizes the velocity variance of all the legs in stance, allowing for smooth acceleration in legged robots. The gait switching generator is modeled as a max-plus linear discrete event system which is translated to continuous time via a reference trajectory generator.

I. INTRODUCTION

Gait generation for many legged robots via central pattern generators (CPG’s) is a well research topic in the robotics community (see Holmes et al. [1] for an extensive review on the elements of dynamic legged locomotion). However, gait transitions are much less explored. Most results are based on continuous-time dynamics, including [2]–[5] or focus on gait energetics to define gait switching points [6], [7]. Most of the methods presented offer complex solutions based on search techniques to find optimal gait transitions. By taking advantage of a class of max-plus linear discrete event systems for modeling legged locomotion, introduced by the authors in [8], [9], optimal gait switching can be accomplished naturally by switching particular system matrices in a max-plus linear system.

Max-plus linear discrete-event systems (MPL-DES) are a subclass of timed DES (classes of discrete event systems where there exists an underlying time structure) that can be framed in systems of linear equations in the max-plus algebra [10]–[12]. DES that enforce synchronization can be modeled in this framework. MPL systems 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. At the time of writing, the theory of max-plus algebras and in Section III we demonstrate how legged locomotion can be modeled by max-plus linear systems. In Section IV we demonstrate how to choose the proper switching gaits to obtain optimal switching in terms of minimizing the ground leg velocity variance, in Section V we introduce transition gaits to enforce constant leg ground velocity, and finally in Section VI we present a variable velocity gait generator.

II. MAX-PLUS ALGEBRA

Max-plus algebras were introduced in the sixties by Giffler [19] and Cuninghame-Green [20]. In the late seventies the second author wrote the first book [10], and in the eighties Cohen et al. [21] presented a system-theoretic view. A few additional books have been published on the topic including [11], [12]. For a historical overview see [22]. The structure of the max-plus algebra is as follows: let \( \varepsilon := -\infty \), \( e := 0 \), and \( \mathbb{R}_\text{max} = \mathbb{R} \cup \{\varepsilon\} \). Define the operations \( \oplus, \otimes : \mathbb{R}_\text{max} \times \mathbb{R}_\text{max} \rightarrow \mathbb{R}_\text{max} \) by:

\[
\begin{align*}
x \oplus y & := \max(x, y) \\
x \otimes y & := x + y
\end{align*}
\]

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

Theorem 1: [11] The max-plus algebra \( \mathbb{R}_\text{max} \) 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 ‘×’ 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 \( \mathbb{R}_{\max} \). Define the matrix sum \( \oplus \), matrix product \( \circ \), and matrix power operations by:

\[
[A \oplus B]_{ij} = a_{ij} \oplus b_{ij} := \max(a_{ij}, b_{ij})
\]

\[
[A \circ C]_{ij} = \bigoplus_{k=1}^{m} a_{ik} \circ c_{kj} := \max_{k=1, \ldots, m} (a_{ik} + c_{kj})
\]

\[
D^\circ k := D \circ D \circ \cdots \circ D_k
\]

where \( A, B \in \mathbb{R}^{n \times m}_{\max}, C \in \mathbb{R}^{m \times p}_{\max}, D \in \mathbb{R}^{n \times n}_{\max} \), and the \( i,j \) element of \( A \) is denoted by \( a_{ij} = [A]_{ij} \). In this context, the max-plus zero \( \mathcal{E} \), and identity \( E \) matrices are defined by:

\[
[\mathcal{E}]_{ij} = \epsilon; \quad [E]_{ij} = \begin{cases} 
\epsilon & \text{if } i = j \\
\epsilon & \text{otherwise.}
\end{cases}
\]

We use the following notation to illustrate the dimensions of the previous matrices:

\[
\mathcal{E}^{n \times n}_{\max} \in \mathbb{R}^{n \times n}_{\max}; \quad E_n \in \mathbb{R}^{n \times n}_{\max}
\]

Finally, we define \( D^0 := E \) and \( x^0 := e \). Now let

\[
A^* := \bigoplus_{k=0}^{\infty} A^k.
\]

If \( A^* \) exists then the vector \( x = A^* \circ b \) solves the system of max-plus linear equations

\[
x = A \circ x \circ b,
\]

with \( A \in \mathbb{R}^{n \times n}_{\max} \) and \( b \in \mathbb{R}^{n}_{\max} \) (see [11], Theorem 3.17). The matrix \( D \in \mathbb{R}^{n \times n}_{\max} \) is called nilpotent if

\[
\exists k < \infty, \forall p > k : D^\circ p = \mathcal{E}
\]

It is always the case that if \( D \) is nilpotent then \( k < n \).

**Definition 2:** The (square) matrix \( A \) is called irreducible if no permutation matrix \( B \) exists such that the matrix \( \bar{A} \), defined by

\[
\bar{A} = B^T \circ A \circ B,
\]

has an upper triangular block structure (an alternative definition states that a matrix \( A \) is irreducible if its communication graph is strongly connected [12]).

Eigenvectors \( \lambda \) and eigenvalues \( v \) are defined in the same way as in the traditional algebra, where \( v \neq \mathcal{E} \):

\[
A \circ v = \lambda \circ v
\]

For max-plus linear systems the eigenvalue of the system matrix represents the total cycle time, whereas the eigenvector dictates the steady-state behavior.

**Theorem 2:** [11] If \( A \) is irreducible, there exists one and only one eigenvalue (but possibly several eigenvectors).

## III. Modeling Legged Locomotion

We model legged locomotion by abstracting the continuous-time motion of the legs into discrete-event cycles. 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 event index. Here, \( k \) is considered to be a global event counter. 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 by \( \tau_f \)):

\[
t_i(k) = l_i(k) + \tau_f
\]

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

\[
l_i(k) = t_i(k) + \tau_g,
\]

where \( \tau_g \) is the stance time and \( t_i \) uses the previous event index 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 \( \tau_{\Delta} \) seconds after leg \( j \) has touched down \( \tau_{\Delta} \) is the double stance time). One can then write the relation:

\[
l_i(k) = \max(t_i(k-1) + \tau_g, t_j(k) + \tau_{\Delta} + \tau_g)
\]

Equation (4) enforces simultaneously that both the leg \( i \) stays at least \( \tau_g \) seconds in stance and will only lift off at least \( \tau_{\Delta} \) 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.

For an \( n \)-legged robot, let the full discrete-event state vector be defined by:

\[
x(k) = [l_1(k) \cdots l_n(k) \cdots t_1(k) \cdots t_n(k)]^T.
\]

The 2\( n \)-dimensional system equations for the cycles represented by equations (2),(3) take the form:

\[
\begin{bmatrix}
l(k) \\
\tau_f(k)
\end{bmatrix} = \begin{bmatrix}
\mathcal{E} & \tau_f \circ E \\
\mathcal{E} & \mathcal{E}
\end{bmatrix} \begin{bmatrix}
l(k) \\
\tau_g \circ E
\end{bmatrix}
\]

\[
\begin{bmatrix}
l(k-1) \\
\tau_{\Delta}
\end{bmatrix} + \begin{bmatrix}
\mathcal{E} & \mathcal{E} & \tau_f \circ E \\
\mathcal{E} & \mathcal{E} & \mathcal{E}
\end{bmatrix} \begin{bmatrix}
l(k-1) \\
\tau_g \circ E
\end{bmatrix}
\]

\[
\begin{bmatrix}
l(k-1) \\
\tau_f(k)
\end{bmatrix} + \begin{bmatrix}
l(k-1) \\
\tau_g \circ E
\end{bmatrix}
\]

According to (5) all legs follow the same rhythm, i.e. all legs rotate with the same period of at least \( \tau_f + \tau_g \) seconds. The introduction of the extra identity matrices \( E \) in (5) results in the extra trivial constraints \( t(k + 1) \geq t(k) \) and \( l(k + 1) \geq l(k) \). This enforces however, that the resulting system matrix will be irreducible. We assume that all leg synchronizations are achieved by enforcing a relation between the next lift-off time of a leg with the touchdown\(^1 \) time of other legs, as in

\(^{1}\)In practice, touchdown can be detected via touch sensors on the feet or by detecting via encoders that the leg has reached the “phase position” where it is safe to lift the other legs.
equation (4). This assumption is expressed by the additional matrices \( P \) and \( Q \) (that we define next) added to equation (5), resulting in the synchronized system:

\[
\begin{bmatrix}
  t(k) \\
  l(k)
\end{bmatrix}
= \begin{bmatrix}
  \mathcal{E} & \tau_f \otimes \mathcal{E} \\
  \mathcal{P} & \mathcal{E}
\end{bmatrix}
\otimes
\begin{bmatrix}
  t(k) \\
  l(k)
\end{bmatrix}
\oplus
\begin{bmatrix}
  \mathcal{E} & \tau_g \otimes \mathcal{E} \\
  \tau_g \otimes \mathcal{E} + Q
\end{bmatrix}
\otimes
\begin{bmatrix}
  t(k-1) \\
  l(k-1)
\end{bmatrix}
\]

(6)

which one can write using simplified notation as:

\[
x(k) = A_0 \otimes x(k) \otimes A_1 \otimes x(k-1),
\]

(7)

**Lemma 1:** [9] A sufficient condition for \( A_1 \) to exist is that the matrix \( P \) is nilpotent in the max-plus sense. Equation (7) can be written explicitly by

\[
x(k) = A^0 \otimes A_1 \otimes x(k-1)
= \mathcal{A} \otimes x(k-1),
\]

(8)

where \( \mathcal{A} = A^0 \otimes A_1 \) is called the system matrix.

For a robot with \( n \) legs let \( \ell_1, \ldots, \ell_m \) be sets of integers such that

\[
\bigcup_{p=1}^m \ell_p = \{1, \ldots, n\}, \text{ and } \forall i \neq j, \ell_i \cap \ell_j = \emptyset
\]

i.e., each set \( \ell_p \) takes elements of \( \{1, \ldots, m\} \) with no overlap between sets. Define \( \tau_p = \#\ell_p \). We consider that each \( \ell_p \) contains the indices of a set of legs that recirculate simultaneously (i.e. liftoff together). A gait \( \mathcal{G} \) is defined as an ordering relation of groups of legs:\n
\[
\mathcal{G} = \ell_1 < \ell_2 < \cdots < \ell_m
\]

(9)

This ordering relation is interpreted in the following manner: the set of legs indexed by \( \ell_{i+1} \) swings immediately after all the legs \( \ell_i \) have reached stance arriving from their own swing. For example, a trotting gait on a quadruped robot where the legs are sorted as in Figure 1, is represented by:

\[
\{1, 4\} < \{2, 3\}
\]

Given the previous notation, the matrices \( P \) and \( Q \) in equation (6) can be generated by: \( \forall j \in \{1, \ldots, m-1\}, \forall p \in \ell_{j+1}, \forall q \in \ell_j \),

\[
[P]_{p,q} = \tau_\Delta
\]

(10)

and \( \forall p \in \ell_1, \forall q \in \ell_m \),

\[
[Q]_{p,q} = \tau_\Delta
\]

(11)

where all other entries of \( P \) and \( Q \) are \( \varepsilon \).

Let \( \tau_\Delta = \tau_f \otimes \tau_\Delta \) and consider the following assumptions (which are always satisfied in practice):

\begin{enumerate}
  \item \( \tau_\Delta, \tau_f > 0; \quad \tau_\Delta \geq 0 \)
  \item \( \tau_\Delta \otimes \tau_f \leq \tau_\Delta^{\otimes m} \)
\end{enumerate}

**Theorem 3:** [23] If assumptions A1, A2 verify then the matrix \( \mathcal{A} \) defined by equations (8), (6) has a unique eigenvalue \( \lambda = \tau_f^{\otimes m} \) and a unique eigenvector \( v \) (up to scaling factor) defined by

\[
\forall j \in \{1, \ldots, m\}, \forall q \in \ell_j : \quad [v]_q = \tau_\mathcal{G} = \tau_f \otimes \tau_\Delta^{\otimes j-1}
\]

(6)

**IV. OPTIMAL GAIT SWITCHING**

Let the gait space for an \( n \)-legged robot be the set of all system matrices for gaits generated from (9) with equations (6), (8):

\[
\mathcal{A}_n = \{A(1), \ldots, A(j_n)\}
\]

One can write the switching max-plus linear system

\[
x(k+1) = A(\mu(k)) \otimes x(k)
\]

where \( \mu(k) \) is a “switching” integer function. By construction, gait switching is stable, in the sense that for two different gaits that recirculate at most \( s_i \) and \( s_j \) legs, will have at most \( \max(s_i, s_j) \) legs recirculating during the transition between both. For example during the transition between a walk and a trot on a quadruped robot, no more then two legs can recirculate simultaneously (note that since we are not taking into consideration the dynamics of the robot this measure of “stability” applies only to the discrete event supervisory controller). By looking at the definition of a gait in expression (9) it is clear that the size of the gait space \( \mathcal{A} \) is combinatorial in \( n \). However, different representations for a gait as an ordered set of sets can lead to the same exact robot physical motion behavior, as in the following example:

\[
\mathcal{G}_1 = \{1, 2\} < \{3, 4\} < \{5, 6\}
\]

\[
\mathcal{G}_2 = \{5, 6\} < \{1, 2\} < \{3, 4\}
\]

\[
\mathcal{G}_3 = \{4, 3\} < \{6, 5\} < \{2, 1\}
\]

\[
\cdots
\]

The difference relies in the fact that the transition between the above defined gaits and a new different gait, say \( \mathcal{G}_4 = \{3, 4, 6\} < \{1, 2, 5\} \), will result in a different transient behavior, as illustrated in Figure 2. This poses the question of how to optimally switch gaits, in the sense of minimizing the variation of the leg stance velocity during gait switching. For applications of climbing robots [2] it is fundamental that all legs exert the same force on the attaching wall at all times, thus motivating constant foot velocity (assuming same power transmission). The same is valid for walking robots, as different leg velocities can result in turning moments that can make the legged platform unstable. For the \( n \)-legged robot with gaits represented by (9) suppose the gait switching mechanism consists on moving a single leg from one group of legs \( \ell_i \) to a different group of legs \( \ell_j \) with \( 0 < i, j \leq m \). By inspecting the eigenvector, one can observe that the moment that a leg in the set \( \ell_i \) lifts of the ground happens at the time instant

\[
(\tau_f \otimes \tau_\Delta)^{\otimes i},
\]

This definition intentionally does not distinguish from running or walking and does not capture gaits where there are multiplicity of cycles between legs, e.g. one leg recirculates twice in the time another leg recirculates once.
assuming the cycle starts at zero time. Analogously, for a leg in the set \( \ell_j \) we get the lift off time to be:

\[
(t_f \otimes \Delta_i) \oplus j,
\]

Moving a leg from the set \( \ell_i \) to the set \( \ell_j \) results in a change of lift-off time of

\[
(t_f \otimes \Delta) \oplus (j-i)
\]

If \( j > i \), then the switching leg will stay in the ground extra \((t_f \otimes \Delta) \oplus (j-i)\) seconds during the transition to synchronize with the new leg group. This is always the case since the time of flight \( t_f \) is fixed. If \( j < i \) then all the legs in the original group of the switching leg will have their lift off times postponed by \((t_f \otimes \Delta) \oplus (i-j)\) seconds. Thus, the larger the magnitude of \( j - i \) the larger will be the ground time variance during the transition. E.g. the gait transition of

\[
\{1, 2\} \prec \{3, 4\} \prec \{5\}, \{6\} \rightarrow \{1\} \prec \{2, 3, 4\} \prec \{5\} \prec \{6\}
\]

has less ground time variance then the transition

\[
\{1, 2\} \prec \{3, 4\} \prec \{5\}, \{6\} \rightarrow \{1\} \prec \{3, 4\} \prec \{2, 5\} \prec \{6\}
\]

The same is true when changing the number of leg groups, e.g. the gait transition of

\[
\{1, 2, 3\} \prec \{4, 5, 6\} \rightarrow \{1, 2\} \prec \{3, 4\} \prec \{5, 6\}
\]

has less ground time variance then the transition

\[
\{1, 2, 3\} \prec \{4, 5, 6\} \rightarrow \{5, 6\} \prec \{1, 2\} \prec \{3, 4\}
\]

This provides a simple mechanisms for choosing gaits without requiring to search the gait space for all structurally equivalent gaits. Figure 3 illustrates the comparison of a non-optimal gait switch a) with an optimal b). To quantify the quality of a gait transition, we introduce the following measure:

\[
\sigma = \frac{\sigma(t_{g1}, \ldots, t_{gn})}{\tau_g},
\]

where \( \tau_g \) represents the true time leg \( g \) is in stance during the transition and \( \sigma \) is the standard deviation. We divide the standard deviation by the desired stance time \( \tau_g \) to obtain an non-dimensional measure. If \( \sigma = 0 \) then the transition maintains a constant stance time for all legs. Note that minimizing \( \sigma \) results in minimizing the variance of the foot velocities during stance (assuming a constant foot velocity for the stance phase range).

V. MULTIPLE \( \tau_f \)-MODEL FOR SWITCHING GAITS

As shown before, by selecting the legs indices in the proper way when switching a gait, one can achieve a better switching behavior. However, by construction, since the synchronization happens at the lift off time, during gait transitions some legs will inevitably stay longer on the ground, which can cause instabilities to the robotic platform. We now show that by manipulating the flight time of each leg independently one can achieve a unique stance time for all legs under well defined assumptions. Consider the new model:

\[
\begin{bmatrix}
t(k) \\
l(k)
\end{bmatrix} = \begin{bmatrix}
E & R \\
R & \tau_g \otimes E \oplus Q
\end{bmatrix} \begin{bmatrix}
t(k) \\
l(k)
\end{bmatrix} \oplus \begin{bmatrix}
t(k-l) \\
l(k-l)
\end{bmatrix}
\]

where the diagonal matrix \( R \) represent different flight times:

\[
R = \begin{bmatrix}
\tau_{1} & \varepsilon & \cdots & \varepsilon \\
\varepsilon & \tau_{2} & \cdots & \varepsilon \\
\varepsilon & \cdots & \cdots & \varepsilon \\
\varepsilon & \cdots & \cdots & \tau_{fn}
\end{bmatrix}
\]

We say that the system matrix of system (6) is parameterized by:

\[
A(G, t_f \otimes E, \tau_g, \Delta)
\]

and that the system matrix of system (13) is parameterized by:

\[
A(G, R, \tau_g, \Delta)
\]

Now consider two different gaits \( G_1 \) and \( G_2 \) with respective eigenvectors \( v_{G_1} = [l_{G_1}, l_{G_1}]^T \) and \( v_{G_2} = [l_{G_2}, l_{G_2}]^T \). During a transition from gait \( G_1 \) to gait \( G_2 \) the extra time each leg will stay on the ground can be computed by:

\[
\tau_{extra} = (l_{G_2} - l_{G_1}) - \min(l_{G_2} - l_{G_1}) \tag{14}
\]

A transition system matrix \( A(G_1, R_1, \tau_g, \Delta) \) can be constructed such that the “extra time” \( \tau_{extra} \) is subtracted from the flight time \( t_f \) so that in the next cycle, now using gait \( G_2 \), will make all \( \tau_g \) the same for each leg. Note that this is only possible if

\[
\tau_{fG_1} \geq \max(\tau_{extra})
\]

If that is not the case, then an additional transition matrix, now using gait \( G_2 \), can be constructed as \( A(G_2, R_2, \tau_g, \Delta) \) such that the time that cannot be subtracted from the transition matrix \( R_1 \) is subtracted from the matrix \( R_2 \). The resulting transition algorithm is summarized as follows:
1) Given two gaits $G_1$ and $G_2$ compute $\tau_{extra}$ via equation (14).

2) if $\tau_{fG_1} \geq \max(\tau_{extra})$ then compute the vector:

$$\tau_i = [(\tau_{fG_1} - [\tau_{extra}]_1) \cdots (\tau_{fG_1} - [\tau_{extra}]_n)]^T$$

and the system matrix

$$A(G_1, \text{diag}(\tau_{i1}), \tau_{gG_1}, \tau_{DG_1})$$

where diag represents the diagonal matrix. The transition sequence is obtained by the following sequence of system matrices:

$$A(G_1, \tau_{fG_1} \otimes E, \tau_{gG_1}, \tau_{DG_1})$$

$$\vdots$$

$$A(G_1, \tau_{fG_1} \otimes E, \tau_{gG_1}, \tau_{DG_1})$$

3) if $\tau_{fG_1} < \max(\tau_{extra})$ then create two transition matrices:

$$A(G_1, \text{diag}(\tau_{i1}), \tau_{gG_1}, \tau_{DG_1})$$

and

$$A(G_2, \text{diag}(\tau_{i2}), \tau_{gG_2}, \tau_{DG_2})$$

where

$$[\tau_{i1}]_i = \max(\min([\tau_{extra}]_i, \tau_{fG_1}), \tau_{fmin})$$

with $\tau_{fmin} > 0$ the minimum leg recirculating time, and

$$[\tau_{i2}]_i = [\tau_{fG_2} - ([\tau_{i1}]_i - [\tau_{extra}]_i) - \min(\tau_{i1} - \tau_{extra})$$

The transition sequence is obtained by the following sequence of system matrices:

$$A(G_1, \tau_{fG_1} \otimes E, \tau_{gG_1}, \tau_{DG_1})$$

$$\vdots$$

$$A(G_2, \tau_{fG_2} \otimes E, \tau_{gG_2}, \tau_{DG_2})$$

Figure 3.c) illustrates an example transition with constant ground times $\tau_g$ and different $\tau_f$ for each leg during the transitions, highlighted by the blue shades of color.

VI. VARIABLE GROUND VELOCITY

Variable velocity can be achieved by modulating the time $\tau$. As presented in [8], reference leg phases $\theta_{ref}$ are generated by the event time sequence resulting from the evolution of equation (6):

$$\theta_{ref}(\tau) = f(x(k), x(k-1), \ldots, x(k-p), \tau).$$

The event state vectors $x(k), x(k-1), \ldots, x(k-p)$ are chosen such that

$$\max(x(k-p)) < \tau < \min(x(k))$$

For the case of the robots illustrated in Figure 1 the function $f$ generates a continuous piecewise constant velocity reference phase. Acceleration is straightforward to achieve by introducing a time modulating function $\alpha(\tau)$ to obtain a new reference phase generator:

$$\theta_{ref}(\tau) = f(x(k), x(k-1), \ldots, x(k-p), \alpha(\tau)),$$

with $\max(x(k-p)) < \alpha(\tau) < \min(x(k))$. A constant accelerating robot can be obtained by choosing $\alpha(\tau) = \alpha \tau$ where $\alpha$ is the desired acceleration. Figure 3.d) illustrates an accelerating gait, with gait transitions for a hexapod robot.

VII. CONCLUSIONS

Modeling via switching max-plus linear systems simplifies the synthesis of optimal gait switching supervisory controllers. Under well defined assumptions it is possible to switch gaits while maintaining the same stance time for all legs, allowing for smooth acceleration.

REFERENCES

Fig. 3. Various gait generation simulations for an hexapod robot with transitions indicated by the blue shades of color. A) Non optimal gait switching for transitions \{1\} ≺ \{4\} ≺ \{5\} ≺ \{2\} ≺ \{3\} ≺ \{6\} → \{5, 2\} ≺ \{3, 6\} → \{1, 4\} → \{2, 3, 6\} ≺ \{1, 4, 5\}. Non-Dimensional standard deviation for transition 1 is \(\bar{\sigma}_1 = 0.57\), for transition 2 is \(\bar{\sigma}_2 = 0.45\), and for transition 3 is \(\bar{\sigma}_3 = 0.80\). B) Optimal gait transitions with fixed \(\tau_f\) for the gait transitions \{1\} ≺ \{4\} ≺ \{5\} ≺ \{2\} ≺ \{3\} ≺ \{6\} → \{1, 4\} → \{5, 2\} ≺ \{3, 6\} → \{1, 4, 5\} ≺ \{2, 3, 6\}, \(\bar{\sigma}_1 = 0.14\), \(\bar{\sigma}_2 = 0.33\), and \(\bar{\sigma}_3 = 0.19\). C) Optimal gait switch with transitions with variable \(\tau_f\), \(\bar{\sigma}_1 = \bar{\sigma}_2 = \bar{\sigma}_3 = 0\). D) Gait transitions with constant acceleration.