The problem of locomotion has been widely studied [?] throughout the past decades. The interest on this topic initially arose from the field of biology, achieving great success on understanding how animals such as mammals and insects manage to coordinate in a natural and effortless way the movement of their legs in order to perform specific gaits corresponding to the conditions they face on a daily basis. It was discovered, that complex networks of neurons are in charge of generating the signals to send to each walking limb, using advantageously the periodic nature of the process. This structures of neurons are called CPG, and the way the operate

EXPLAIN HERE CENTRAL PATTERN GENERATORS

Central pattern generators do not actuate each limb independently or are based on the senses of animals/humans, this is because in most of the situation, walking constitutes a periodic activity, where the legs move following the same patterns with phase differences (which change depending on the type of gait that being used). Depending on the situation that the agent is facing, it naturally changes the parameters of the gait (such as the time that the legs remain on the ground or the air or the phase differences between each leg) or even changes the gait completely (such as going from walking to running).

In robotics, this type of structures have been emulated by using simplified models of the neurons and their interactions. The simplest example of how central pattern generators are implemented in robotics

EXPLAIN CPG IN ROBOTICS

Using this kind of framework, various types of gaits can be achieved. They way that CPG's are used differs among all different platforms. Some examples [?] do not use any kind of feedback control in order to generate gaits, although the main purpose of this kind of study is to analyze the animal behavior using robotic models to simulate gait generation. On the other hand, CPG's can be used to effectively generate angle references for each of the legs in robotic platforms [?], to be tracked via control strategies (the most widely used in humanoid and animal-like platforms is computed-torque control and feedback linearization techniques).

Despite the elegance that locomotion generation using CPGs displays, roboticists are facing big challenges regarding robustness against unstructured terrain and dealing with disturbances. In most cases, when encountering unfavorable situations, the use of proprioceptive (information coming from the agent itself) and exteroceptive (information coming from the environment) sensory information in order to change the parameters of the gait such as coupling variables, ground and flight times and velocities of each leg, among others. This framework has proven to efficiently deal with complex types of terrain. In some specific cases, gait switching has been achieved [?], although no control is implemented in the locomotion planning routines.

It can be seen that even in the simplest case, a set of nonlinear differential oscillator equations describe the evolution in time of each limb. This is due to the intrinsic periodicity of the process. This provides an extra difficulty in order to change from one gait to another, especially as the number of legs increases. Another way to approach the locomotion planning problem, is to model it via Discrete Event Systems [?]. In recent years the use of tools such as petri nets and max-plus algebra system have been tested and proved to generate locomotion patterns. Regarding the latter one, max-plus algebra systems have provided not only a simpler and systematic way to generate gait motions, but also due to the its linear (in max-plus) properties provides tools to analyze transitions and velocity of the generated planning.

Max-plus algebra is a kind of "tropical algebra" that complies with the axioms of an algebraic ring [?]. The algebra is defined by:

$$(\Re_{max}, \oplus, \otimes, \varepsilon, e) \tag{1}$$

where:

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

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 of two numbers, 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:

$$(\Re_{max}^{n\times m}, \oplus, \otimes, \mathcal{E}, E) \tag{2}$$

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:

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

Powers of matrices can also be defined as:

$$D^{\otimes k} := D \otimes D \otimes \ldots \otimes D. \tag{3}$$

The max-plus algebra structure corresponds to a commutative idempotent semiring. Max-plus algebra represents a very useful tool to model timed events such as railway scheduling [?] or traffic control [?]. Robot locomotion can be modeled as well as a discrete event system, making use of a subclass of petri nets called timed event graphs [?]. Subsequently, one can represent the evolution equations for a subset of timed event graphs as max-plus algebra linear systems. Gaits can be represented and controlled via switching max-plus linear systems of the form:

$$x(k+1) = A \otimes x(k) \tag{4}$$

In the case of locomotion the parameters and states involved in the process are given as:

- i Index to indicate each of the legs.
- $t_i(k)$ Touchdown time of leg i.
- $l_i(k)$ Lift-off time of leg i.
- τ Current time instant.
- τ_f Time leg spends in flight (swing).
- τ_g Time leg spends on the ground (stance).
- τ_Δ Double stance time (this can be adjusted depending on the gait that it is desired to achieve).

Using this parameters one could write the equations of a leg cycle as:

$$t_i(k+1) = l_i(k+1) + \tau_f \tag{5}$$
$$l_i(k+1) = t_i(k) + \tau_g \tag{6}$$

Then if the state vector $x(k)$ is defined as:

$$x(k) = [\underbrace{t_1(k) \ \ldots \ t_n(k)}_{\textbf{touchdown}} \ \underbrace{l_1(k) \ \ldots \ l_n(k)}_{\textbf{lift-off}}]. \tag{7}$$