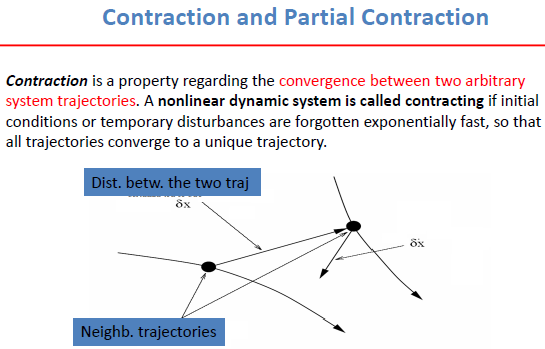
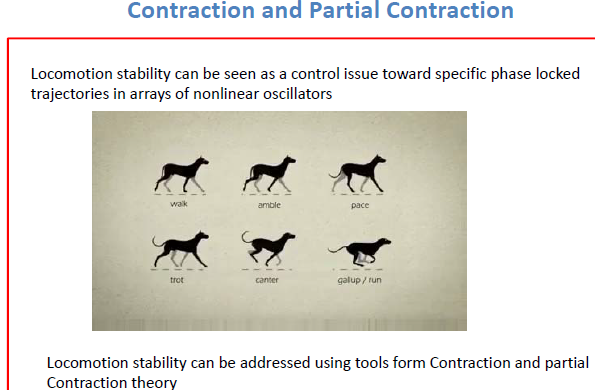
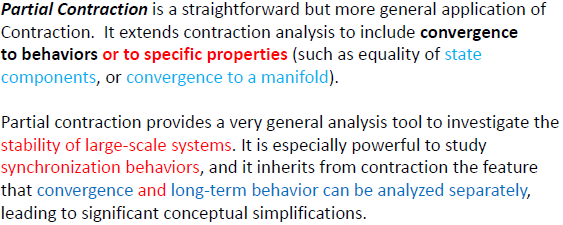
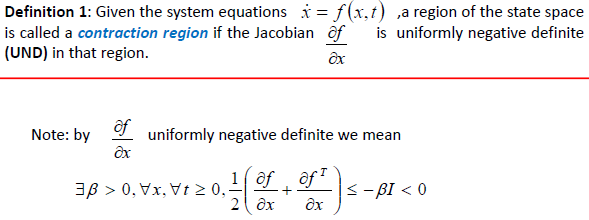
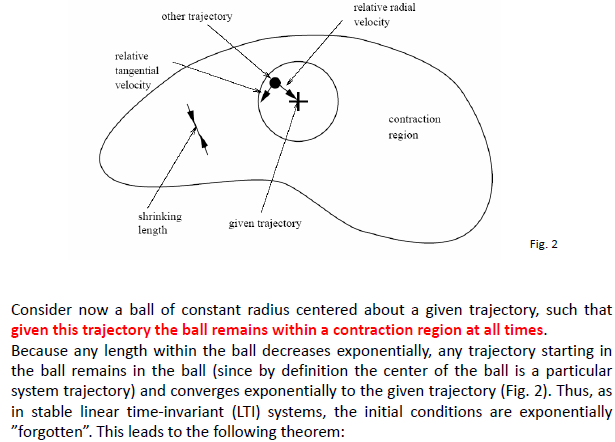
Lesson 14

In this pic, from a video, we can see a different gait for quadruplets. Can be studied not only for the quadruplets, but also for hexapods and so on. It is important to understand in which framework to study these behaviours, so as to ensure some results on stability, feasibility. In particular in order to have stability, it is important to have an alternation of stages of the legs. In fact, there is a perfect alternation between the phases of the diagonal legs of the quadruped. So, one changing between one Patter locomotion respects another one, there is something like a motion respecting a different basin of attraction, so we can think at different gaits that are controlled by different basins of attraction of our neural network. Thanks, the framework of the reaction-diffusion cellular nonlinear network, we can obtain different gaits by changing the connections or the arrangement of the connections between cells and legs. Now, we will see how this reaction- diffusion system can be formally analysed under the theory of ***Contraction and Partial Contraction framework***. Indeed, now we have an analytic framework that guarantees the analytic condition under which we can have stability in the locomotion gaits. So, a stable attractor for the overall non-linear system that contains a large amount of neurons together should be a large amount of basin of attraction, but in some sense we have to guarantee the gaits are governed by the unique attractor for all huge phase space. We will see some analytic tools, but our focus is to choose the parameter for our reaction-diffusion system in a such way to guarantee that when we select the parameter to obtain the walk, amble and so on, these will be the overall attractor for the overall network with a precise gait. We know that the reaction diffusion law is sufficient in order to obtain a particular gait, but from the implementations point of view there will be some spurious basin of attraction that is not covered by the law, but thanks to this new framework we can guarantee the unique solution.

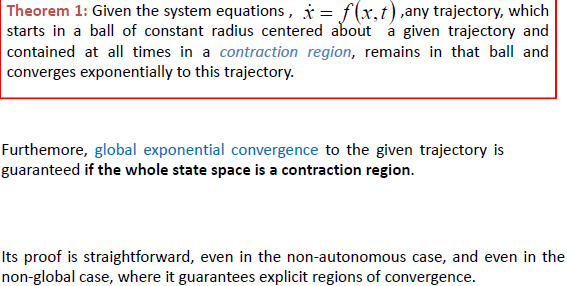
Let us see, two different trajectories. We consider the trajectories as nominal, as if they were pieces of a limit cycle, and we consider the distance between the two trajectories as disturbance and a as a gradient in time. Through the contraction property we can consider that the disorder will be lost exponentially fast so as to have a single trajectory. This means that any disturbance will be lost, and any close trajectory will collapse into the nominal one, is a concept of stability, if we want to do a parallelism with the Lyapunov stability, while the latter guarantees the collapse at an equilibrium trajectory here we guarantee the collapse in an equilibrium trajectory. So, in general, I no longer want an equilibrium point, but I would like a steady state limit cycle which could be my nominal trajectory in such a way as to have the collapse of all trajectories in one round of latter.

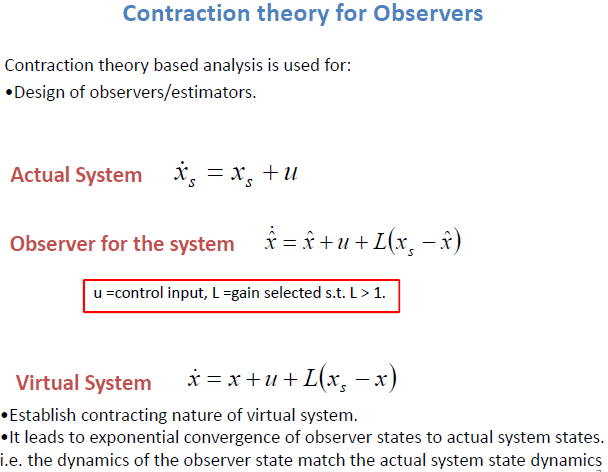
“To Specific properties" means a limit cycle, as mentioned above, or a phase shift limit cycle, for example. "Stability of large-scale systems" means that for example in our case we have many neurons with 2-3 state variables that lead to different behaviours. 

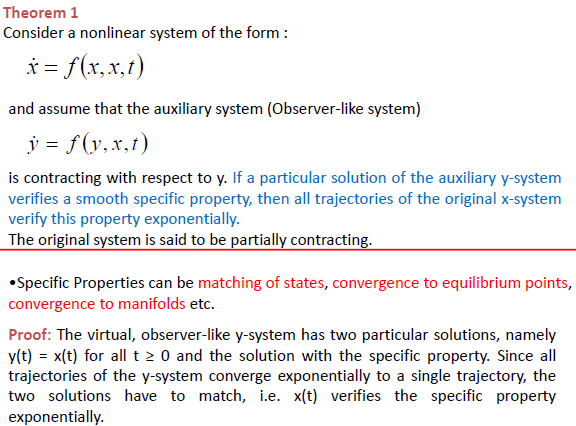
So, let us see some definitions:

🡺 Beta is a constant, I is an identity matrix. We are looking for the Jacobian, what does it mean ? We are looking for the incremental motion even if the system is nonlinear, we will impose some constraints for the incremental motion such that the results gained from the incremental motion could be extended for the overall contraction region that should be larger respect the classical Lyapunov region that should be as small as possible. 

For example, this could be the region where we are talking until now. We can see that the given trajectory [sign +] is coming out of the screen towards us. The black ball is as if it were another trajectory caused by a perturbation, characterized by a radial and tangential velocity, the latter can be dangerous because, if it were too high, the trajectory could escape from the ball. From the text it is understood that the radial velocity, which converges exponentially, must be faster than the tangential velocity, which can bring the trajectory out of the ball, so as to facilitate the collapse towards the given trajectory.

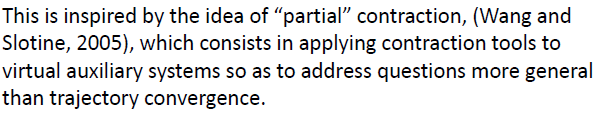
These concepts lead at the theorem, at the left. The last sentence means, if we do not reach the global exponential convergence, we can guarantee the convergence to a particular contraction region.

Now, let us see the parallelism of the contraction theory recalling the theory of observer. Let us consider the contraction theory from the point of view of the observer. The contractive system we want to build is similar to the construction of an observer. Consider that we have an system that is a simple integrator plus an input signal. Remember that the observer is equal to the original system less than an L matrix that looks at the valuation error. Typically, L > 1 to have the convergence. We create the virtual system that is similar to the observer, where the variable of the system has been substituted with that of the perturbation. So, the virtual system is the same as the observer. In fact, if the system is observable it is possible to see the evolution of the system, and if the observability is satisfied I can choose a gain L such that xs -x is equal to zero, so the solution will be robust against the perturbations.

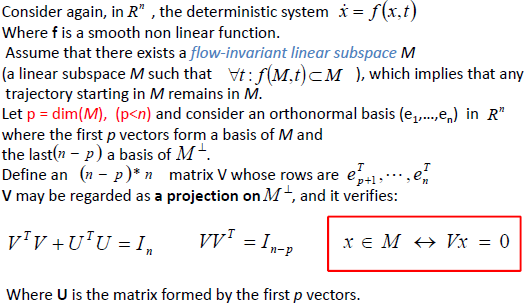
Until now, we have seen the general idea of the *contraction*, now we will see the *partial contraction*. We consider that we have a hundred neurons, where each of these will have two state variables, each state variable will have a trajectory and we want to establish a connection between these. I mean, I want the state variables to have some specific properties. That is, I would like some constraints on some specific state variables that are a partial contraction.

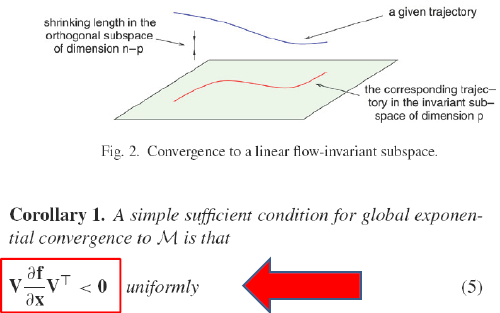
So, if I am interested in a particular solution, for example some phase relationship between the state variable and if the latter are not contradictory to this property "f()" [example phase shift], then all state variables of the original system will obey this. Referring to the test: if y(t) obeys the contraction property then it will obey x(t).

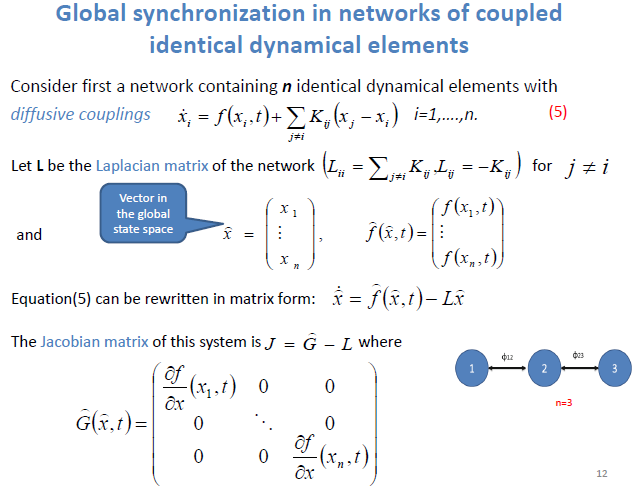


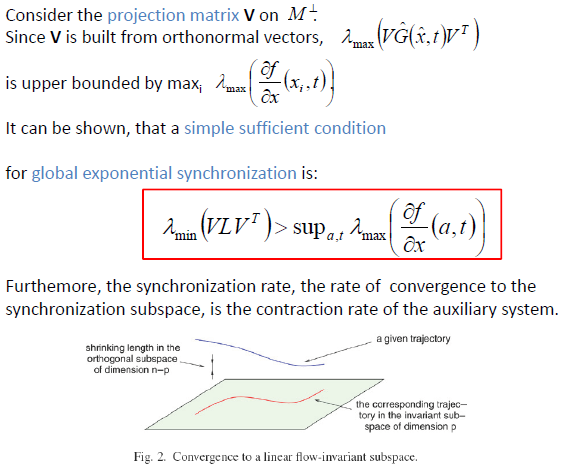
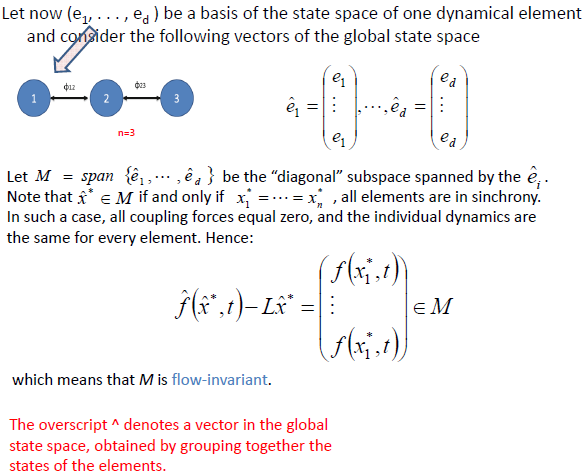
Now, how can this property be defined? It can be defined from the definition of “ *flow-invariant subspace*”. 

Let us see with more details.

So, for every x defined in M they will be contained in M and therefore every trajectory that starts from M will remain in M. Now let us analyse the dimension in M. So, for every x defined in M they will be contained in M and therefore every trajectory that starts from M will remain in M. Now we analyse the dimension in M. p < n means that f(M,t) i.e., M is a subspace of Rn. After the middle passages in the theorem, let us define V matrix, which has n – p rows and n columns which lines are the transpose of the p+1 to n lines that generate the remaining lines to complete the base. We build last equation, and if x M of course Vx = 0 because V is the orthonormal subspace respect M. So, if we can build the orthonormal subspace then a simple condition for global exponential convergence at M is: 

After this explanation, let us think about the geometric idea. Let us imagine in blue the given trajectory of the original system and we defined the *flow-invariant subspace* in red, the latter is the projection of the report which I would like to see satisfied by the original system. I would like the blue one to be trapped in the red one and a sufficient condition is that (5) is fulfilled. 

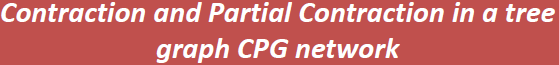
Let us see which is the Jacobian in the global system, because now our system typically includes an equal number of neurons distributed in space. In our case there are 3 neurons that are identical dynamical elements. f(,t) is the nonlinear part of the system. In this case we will have , , that represent the dynamic of each neuron, with 2-3 states variables each. The nonlinear part is connected with the diffusive couplings. We can build the Laplacian and after we will define a Vector in the global state space, that contains all the state variables in each neuron and so on, same thing for the nonlinear part. So, we can rewrite in another form, as we can see in the pic, and finally we compute the Jacobian that is equal to the Jacobian in the diagonal part minus the Laplacian.

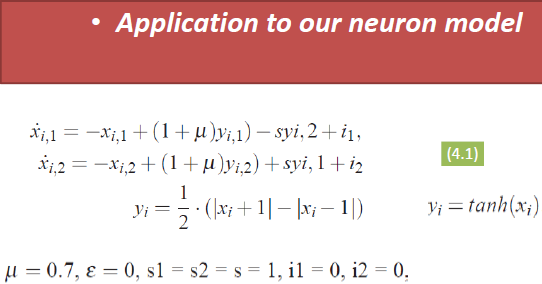
Of course, the neurons are systems with a dynamic of second order therefore (e1, e2). , are column vectors that contain respectively the first and second state variable of each neuron. Which is the main idea ? If we look the (5) and when the system reaches the trajectory which is nominal defined by the nonlinear part which is autonomous the error at right is equal to zero, and this error will act as a feedback for the system; moreover, when the error is zero means that I have no flux between two state variables and better between two neurons and they will obey at same properties.

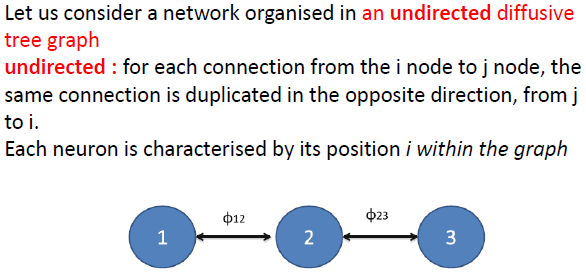
Now, let us introduce the main results in the red box. Of course, G represents the Jacobian that is the disturbance, and it represents how big is the effect of disturbance, since V is built by orthonormal vectors the maximum lambda of the part that contains G is upper bounded by the maximum lambda of the Jacobian value for each state variable flow on the trajectory. At the end can be demonstrated that the minimum eigenvalues of the orthonormal projection Laplacian on the orthonormal subspace should be larger than the superior extremum of lambda max of the Jacobian matrix of a,t. I put a, instead of x, because it is the variation of the x state variable all along the nominal trajectory that I would be the attractor set of points.

From the practical point of view: in any point in the phase plane, containing the limit cycle, I will calculate the Jacobian, after I will calculate the lambda max and I will calculate the superior of lambda max, that will be a number, if the minimum eigenvalue of the projection of the Laplacian matrix will be larger of the “sup”, then I will have a sufficient condition for the ***global exponential synchronization***.

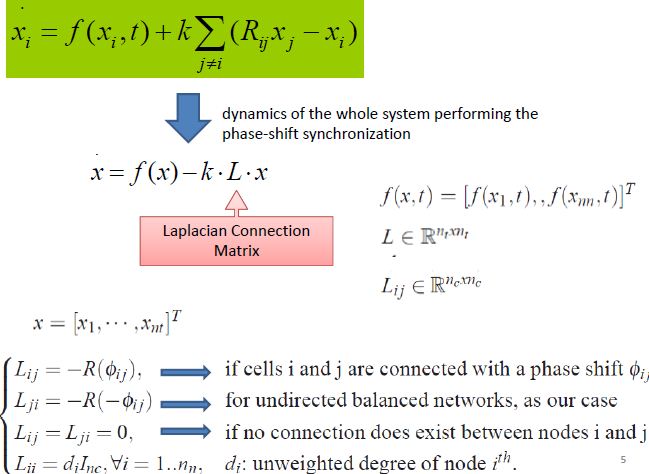
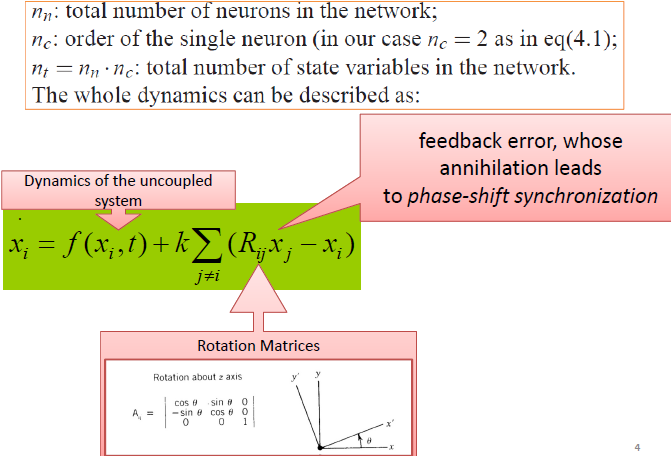
So if the trajectory in red, the flux invariant subspace, and having the projection on M where I defined the relation to which you must obey, and I have V orthogonal projection of M, the formula in the red box tells us that if we calculate the Jacobian of the red trajectory and see its maximum eigenvalue and if I calculate the value of the sup and compare it with the minimum eigenvalue of the orthogonal projection V and the eigenvalue of V is larger, means that the system is sufficiently dissipative to contract the overall phase plane in the one below where the red trajectory is located, making a comparison, , "height" equal to zero. So, at steady state all state variables will obey the property that I defined for the system that are defined in M flow invariant subspace.

🡸 References

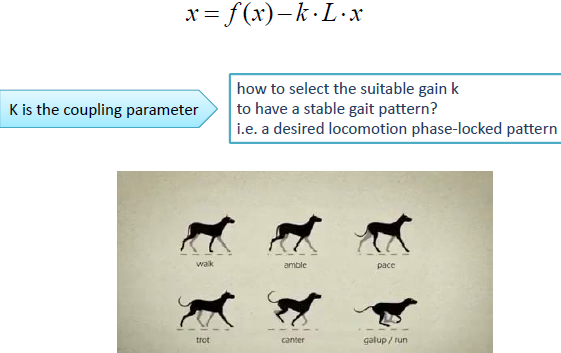
How will this be implemented ? Now, we will go into the application or analysis into the *Contraction and Partial Contraction in a tree graph CPG Network*, tree graph is a graph that does not contain loops. If we have a loop I will have to obey some periodic conditions which do not allow us an arbitrary selection phase relationship between different oscillators. 

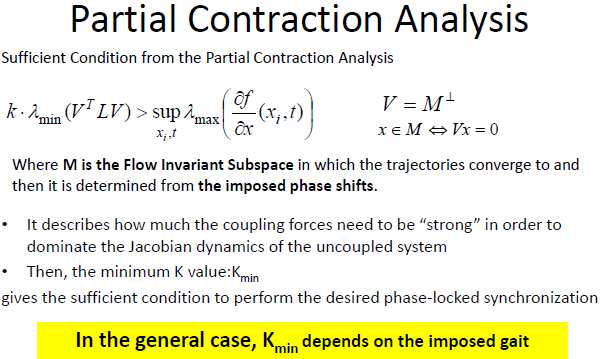
So, let us consider our neuron, that realizes a slow fast limit cycle under the point of view of i1 and i2, if they are not zero. Anyway, we will consider these currents as zero and we will create something like a harmonic oscillator. After we will substitute instead the classical yi (PWL) with a hyperbolic tangent for making the concept of smoothness of the nonlinear function f, that is needed for our implementation.

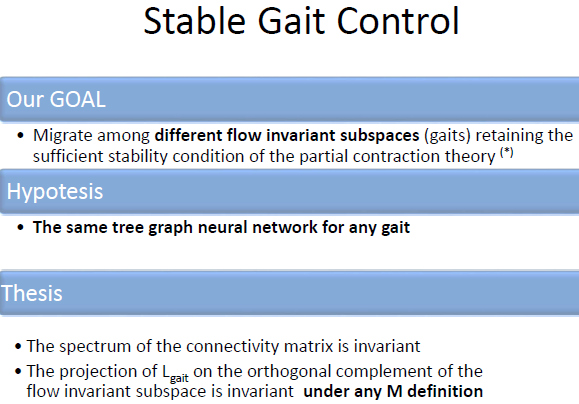
So, let us see the Contraction and Partial Contraction theory implemented in a tree graph CPG Network, that is summed up by the three neurons.

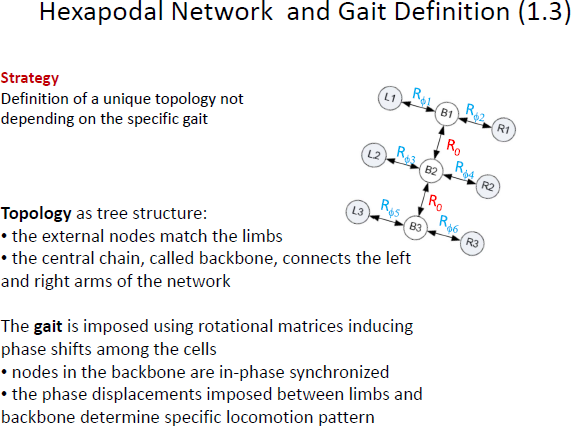
If we look at the equation in green it is an equation that describes the whole dynamics of the network, from the point of view of the Laplacian. Let us analyse the equation. “f(xi,t)” is the dynamic of the single uncoupled system [slow-fast limit cycle], the remaining is part related to the diffusion part, with k the gain. With the rotation matrix a feedback error is formed and if it can have zero I lead it into a phase-shift synchronization. When xj rotated of 30° is equal to xi, both of the corresponding neurons will be phase shifted. I have imposed the converge, if the system will be contractive, to fixed phase shift synchronization for example, synchronization between different legs. I will obtain an efficient locomotion at the steady state.

Let us move in the new equation, that is the compact form. We will remember that “f(x)” is the column vector of all state variables of all cells. We have seen how to define the *Laplacian Matrix*.

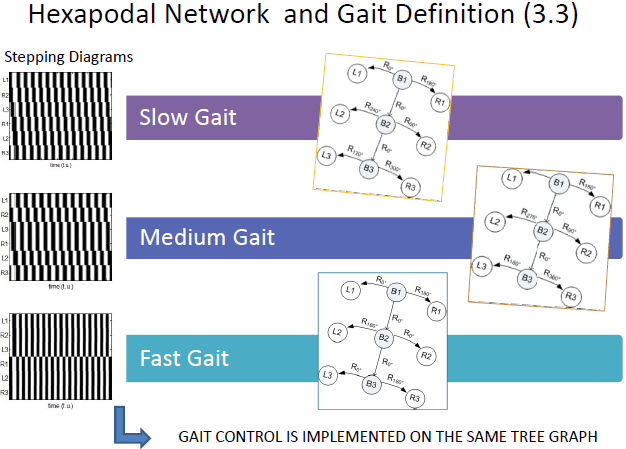
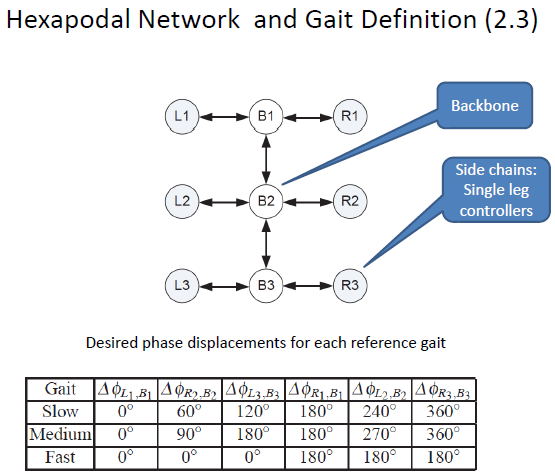
Our task is to implement a particular type of walk, like pace, trot, and so on. The equation governs the phase shift between the legs and allows us to have a particular walk. So, given L, *which is the suitable k that allows to have a particular phase shift synchronized to favourite a particular walk* ?

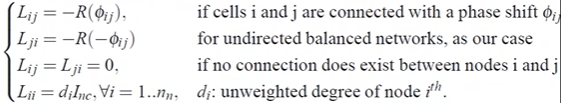
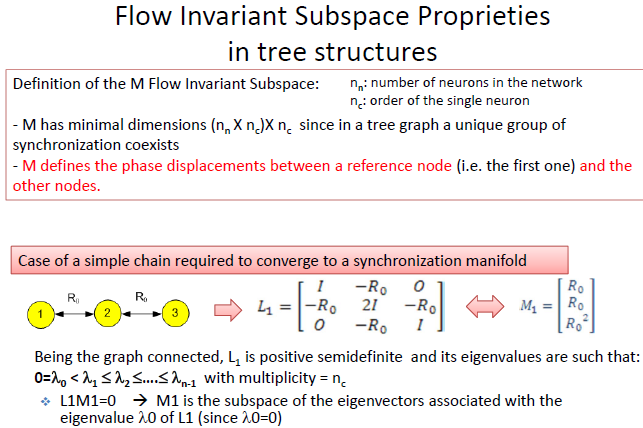


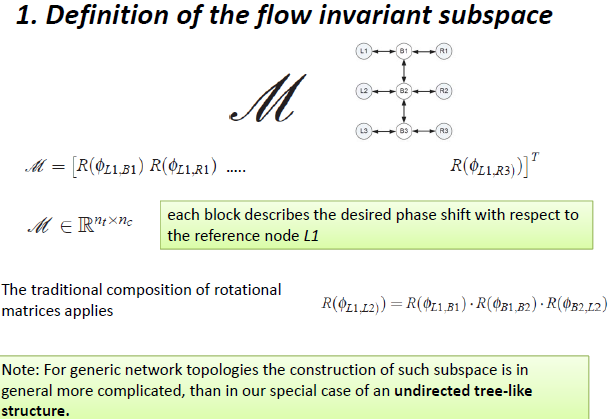
Let us use the results that come from the Partial Contraction analysis. We define a flow-invariant subspace that should contain the phase shift definition among different neurons. When I define M, I will define an orthogonal subspace V and after we will satisfy the equation on the left. So as for k, I will find a k so that the product is greater than the maximum Jacobian eigenvalue. Moreover, a particular gait that is represented by a particular phase shift, that it is possible to define from the imposition of M, will obey that equation. The overall analysis is not complicated because I have the alternation of the reaction and diffusion part. The reaction part is possible to analyse it a priori from the unpaired system because I can analyse the single dynamics of every single neuron. For example, I linearize the system I extract the maximum eigenvalue I calculate the upper part and check with the product if it is satisfied. *Each gait will have the same value as k, so you will find a value that will suit all gaits*, this way each eigenvalue of the L connectivity matrix is invariant for each desired gait.

At left, we have a summary of what has been said. 

Let us see the strategy. We will define a topology as a tree graph. R0 means that we have no phase shift and B1,B2 and B3 are synchronized. Moreover, L1,R2 and L3 are synchronized between them after the remaining legs are shifted 180° in order to have the locomotion.

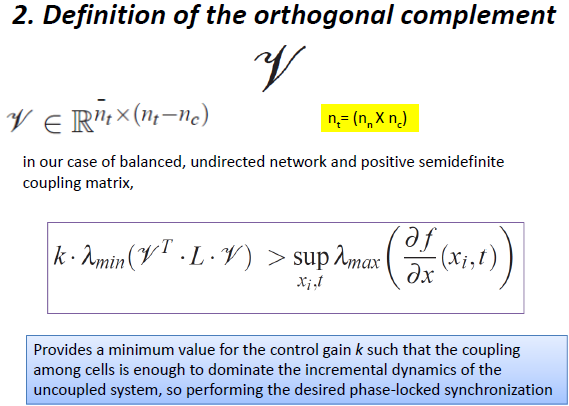
In the table we can see the shift that occurs in the robot.

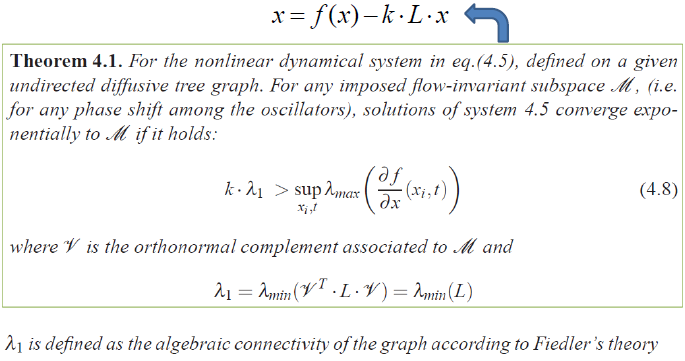
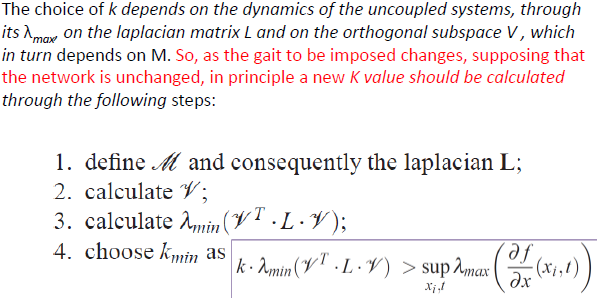
We can see the different stepping Diagrams with the leg synchronized, in accordance with the Rotation Matrix, a group of three as I said before.

How is the Laplacian matrix ? We have to follow the pic above. The first row, second column, represents the rotation from neuron 1 to neuron 2 which is R0 = 0 because there is no shift at this point. The blocks with “zero” means that we have no connection between 1 to 3 and vice versa and so for the remaining parts. Instead M1, that is the flow invariant subspace, is the representation of different neurons that contains the rotation matrices of each neuron compared to the first. In fact, there is the rotation matrix of the first neuron, the second, and the third that is given by the product of the first two in accordance with what was said previously. If a neuron has a second-order dynamic it means that it has a multiplicity of 2, etc. it is a result of the Laplacian of a connected graph. Instead, if I define the first orthogonal column of L1 with respect to the first of M1 its projection will be zero, due to the fact that the first eigenvalue is zero M1 it will be the subspace of the autovector associated to the latter.

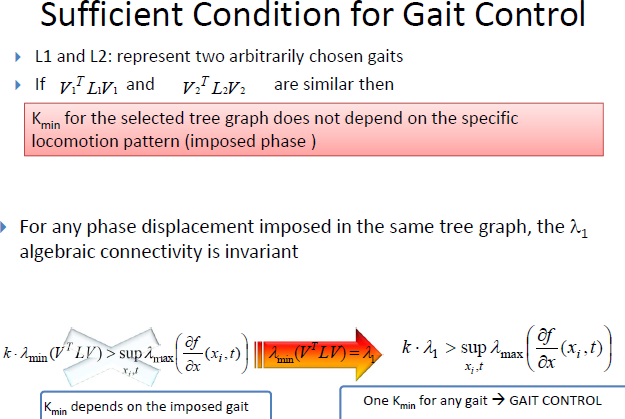
M will be the column arrangement of all the rotation matrices. It will be 9 x 2. In which 2 means the numbers of the state variable for each neuron.

Of course, a particular matrix like in the picture, can be obtained with the product of the other, like in theory.

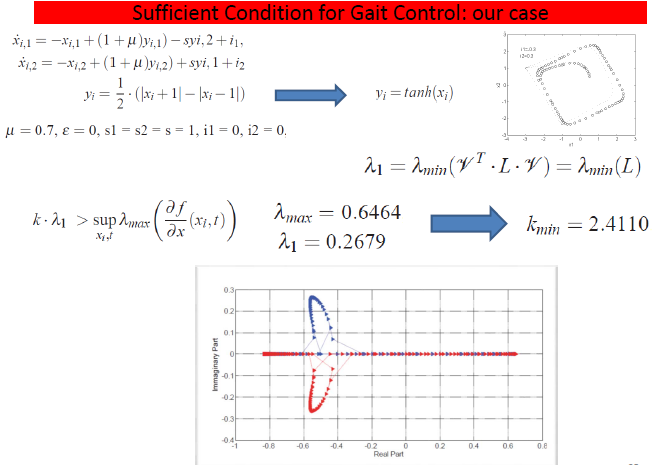
*In this case the graph is like an elastic wire which maintains the tendency to stay in the orthogonal direction respecting the desired property, defined by M, in which the state variable should obey.*

*Indeed, this is the dynamic tendency to escape.*

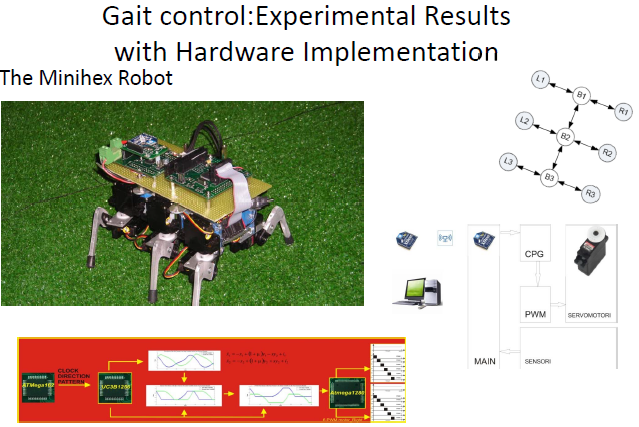
Theorem associated with the previous explanation is not important to study the demonstration.



We can say that if the two projections are similar then the K are the same because there are the same eigenvalues.



This is the numerical result of what explained above. Remember that we made a substitution between two non-linearity [yi], after we calculate the Jacobian Matrix all along the state variables that belong at the limit cycle. I have two state variables, and I will have two lambda max. We have some cases where the real part of lambda max belongs to zero and we could go towards the instability. In order to overcome this, in accordance with the formula we calculate the unit gain k, that for any gaits imposed guarantees sufficient condition for global exponential convergence to this limit cycle with arbitrary phase shift imposed.

**We summarize once fixed the particular phase shift therefore the displacement between the legs, that is the gait, we define the matrix of rotation between a node and another and therefore this part to the Laplacian. After I build the algebraic complement of the Laplacian that is V, I will calculate the minimum eigenvalue of the product between VTLV that will be invariant because L is invariant on arbitrary rotation then on the decoupled starting oscillator I calculate the Jacobian matrix on the limit cycle for all the points then I enrol the maximum eigenvalue and then I calculate the upper extreme and will come a number that will eventually give me the value of k. So, I am going to fix k in such a way that I have strong enough to crush the dynamics of the neuron that tends to run away. So even if this dynamic that wants to escape is unstable, through the feedback on the state [] I crush and constrict it to obey the constraints imposed by M.** 

We can see an experiment of the previous explanation.

