SUPPLEMENTARY INFORMATION

DESIGN AND FABRICATION OF THE SPINAL IMPLANT

Spatial distribution of leg motoneurons

Monkeys (n = 7, **Supplementary Table 1**) received injections of Fastblue (4 % in sterile saline, 1000 µl per muscle) or Fluoro-Ruby (10 % in sterile saline, 1000 µl per muscle) in up to two distinct muscles of the left and right legs, allowing tracing up to four muscles per monkey. The following muscles recorded during locomotion were traced: gluteus medius, iliopsoas, rectus femoris, semitendinosus, gastrocnemius medialis, tibialis anterior, extensor digitorum longus, and flexor hallucis longus. The locations of the retrogradely traced motoneurons were reconstructed in 3D from serial sections of the spinal cord using Neurolucida. We merged the reconstruction from several monkeys into a unified digital library using the morphology of the spinal segments as a landmark.

Anatomy of the vertebrae

Imaging of the vertebrae for the design and visualization of spinal implants was conducted using the micro-computer tomography scanner Skyscan 1076 (Bruker µCT). Scanner settings were adjusted to avoid artefacts induced by the metallic parts of the vertebral orthosis (0.5-1 mm aluminum filter, voltage 70-100 kV, current 100-140 µA, exposure time 120-160 ms, rotation step 0.5 deg). The resulting projection images were reconstructed into 3D renderings using NRecon and GPURecon Server (Bruker µCT). The spinal cords of monkeys were imaged post-mortem, after explant. Segmentation and 3D models were constructed with Amira® (FEI Vizualisation Sciences Group). The 3D shape of vertebrae was derived from this micro-computed tomography imaging. The spinal cords of three monkeys were imaged, and the entire bone structure was reconstructed in 3D. The 3D renderings were exported in the virtual reality modelling language file format WRL that was later merged with spinal tissue and dorsal root reconstructions. Measurements of relationships between vertebra and spinal segment morphologies were performed on fresh tissue. For each subject, the spinal segments were identified based on the innervation of the dorsal roots. The centre of the segment was defined as the entry point of the rootlets. After measuring the length of vertebra, and the relationships between vertebra and spinal segments, the entire spinal cord was extracted, and the roots moved perpendicular to the spinal cord to clearly visualize the segments. The location and length of each segment was then calculated.

Reconstruction of spinal segments and dorsal root trajectories

The spinal cords of three monkeys were dissected, fixated overnight, and transferred to 30% phosphate buffered sucrose for cryoprotection. The dura mater was opened along the rostrocaudal axis, and gently moved on the side. For each spinal segment, the dorsal root ganglions were identified. The corresponding root was retracted cranially and laterally. The entire length of the root was painted, from the cut extremity to the entrance into the spinal segment. All the painted roots from S1 to L1 were repositioned to their original location along spinal segments, and the dura mater was sutured. The spinal cord was then frozen and cut into 80 µm thick slices using a cryostat (Leica Instruments). Reconstructions were conducted using every 4 sections, corresponding to intervals of 320 µm. The slides were assembled into the Neurolucida image analysis software (MBF Bioscience) to reconstruct the color-coded dorsal roots trajectories and lumbosacral segments in 3D (**Extended Data Fig. 1**).

Spinal cord and vertebral morphology

All the 3D reconstructions derived from micro-computed tomography imaging and anatomical experiments were merged to generate a global model including the bony structure of vertebras, the shape of spinal segments, the trajectory of each dorsal root and the anatomical locations of motoneuron columns. This model supported the optimization of the spinal implant and vertebral orthosis.

Polyimide-based spinal implant

We designed and fabricated Polyimide-based epidural spinal implants for monkeys using technology similar to that previously described in rats[1]. Briefly, the spinal implants were fabricated using ultra-violet (UV) photolithographic patterning of a photosensitive Polyimide, as well as micro electroforming to create gold electrodes and embedded gold interconnects. Polyimide is a mechanically and chemically robust polymer material exhibiting a high level of biocompatibility. Processing of the implants was performed on wafer scale, which allowed parallel fabrication of 4 devices or more on a 125 mm silicon wafer serving as carrier wafer. Processing started with deposition of a 20 µm thick Polyimide film by spin coating on the silicon substrate. UV lithography was used to shape the Polyimide-based bottom layer, which constituted the footprint of the implant. A 200 nm thick gold layer was deposited using vacuum evaporation and lithographically structured providing a conductive seed pattern. Gold was then electroplated to a height of approximately 6 µm to create the electrodes and leads. A 20 µm cover layer of photosensitive Polyimide resist was applied by spin coating to uniformly cover the bottom and electrode structure. A final UV lithography step was conducted in order to structure the top layer of the implant, and create openings over the electrodes and contact pads. The contact interface was embedded in a shaft that oriented the electrode implant towards the spinal cord through the laminectomy. 3D micro-computed tomography scans steered the optimization of the size and shape of the vertebral orthosis, which was realized using 3D laser sintering of medical grade titanium. The titanium orthosis enabled a reliable attachment of the implant onto the vertebra. The process flow, shape and dimensions of Polyimide-based electrode implants are detailed in **Extended Data Fig. 1**.

IMPLANTABLE PULSE GENERATOR

The spinal implants were interfaced with an implantable pulse generator commonly used for deep brain stimulation therapies (Activa RC, Medtronic, USA), which supported the delivery of monopolar, constant-current or constant-voltage stimulation protocols using the case of the generator as the anode. To support real-time triggering capabilities, we developed an investigational firmware for the implantable pulse generator that enabled stimulation across multiple sites (**Extended Data Fig. 2**). In addition, we developed the Neural Research Programmer software application that uploaded the stimulation protocols to the pulse generator and relayed stimulation protocol trigger commands from our real-time analysis software to the pulse generator. Trigger commands were transmitted wirelessly over a chain of devices consisting of a Bluetooth-to-infrared module (ACTiSYS Inc.) and a modified Patient's Programmer (Medtronic, USA). Wireless communication between the Control Computer and the implantable pulse generator was reliable over up to 5 m with latencies of about 100ms. These latencies accumulated through the communication chain between the Control Computer and the implantable pulse generator. Distribution of latencies calculated from experimental data are reported for each step of the communication chain (Extended Data Fig. 2). The Neural Research Programmer was accessed from the control computer through the user datagram protocol (UDP). This system is an investigational prototype restricted to research that is not approved for commercial use.

DECODING OF MOTOR STATES FROM NEURAL SIGNALS

Decoder calibration

The extracted gait event timings served to identify sets of neuronal features, representing measurements of foot off and foot strike motor states that were used to calibrate the decoders. Specifically, sets of fo and fs, FO and FS, were used to derive the foot off and foot strike motor state classes of neural features, C_{FO} and C_{FS}:

$$C_{FO} = \left\{ \bar{a}_{FO} \middle| \bar{a}_{FO}(i) = \begin{bmatrix} x_1\left(fo(i) - \Delta t_{FO}\right) \\ x_1\left(fo(i) - \Delta t_{FO} - \Delta t\right) \\ \\ x_1\left(fo(i) - \Delta t_{FO} - \left(N_{TP} - 1\right) \cdot \Delta t\right) \\ x_2\left(fo(i) - \Delta t_{FO}\right) \\ \\ x_{N_{CH}}\left(fo(i) - \Delta t_{FO} - \left(N_{TP} - 1\right) \cdot \Delta t\right) \end{bmatrix} \right\}$$

$$C_{FS} = \left\{ \bar{a}_{FS} \middle| \bar{a}_{FS}(i) = \begin{bmatrix} x_1\left(fs(i) - \Delta t_{FS}\right) \\ x_1\left(fs(i) - \Delta t_{FS} - \Delta t\right) \\ \\ x_1\left(fs(i) - \Delta t_{FS} - \left(N_{TP} - 1\right) \cdot \Delta t\right) \\ x_2\left(fs(i) - \Delta t_{FS}\right) \\ \\ x_{N_{CH}}\left(fs(i) - \Delta t_{FS} - \left(N_{TP} - 1\right) \cdot \Delta t\right) \end{bmatrix} \right\}$$

where $\bar{a}_{FO}$ and $\bar{a}_{FS}$ are feature vector members of classes C_{FO} and C_{FS}; N_{TP}=3 (pre-injury) or 5 (post-injury) is the number of spike rate measurements taken from the same neural channel; Δt=125ms (pre-injury) or 100ms (post-injury) is the temporal difference between the two consecutive spike rate measurements taken from the same neural channel; N_{CH}=96 is the number of neural channels; and Δt_{FO} and Δt_{FS} are the foot off and foot strike temporal offsets (for derivation of offsets, see section “Calculation of temporal offsets”). Another class C_{OTHER} representing states other than foot off and foot strike gait events was formed:

$$C_{OTHER} = \left\{ \bar{a}_O \middle| \bar{a}_O(i) = \begin{bmatrix} x_1(t_i) \\ x_1(t_i - \Delta t) \\ \\ x_1(t_i - (N_{TP} - 1) \cdot \Delta t) \\ x_2(t_i) \\ \\ x_{N_{CH}}(t_i - (N_{TP} - 1) \cdot \Delta t) \end{bmatrix}, \begin{matrix} t_i < fo - \Delta t_{FO} - 20ms \\ t_i > fo - \Delta t_{FO} + 20ms \ \ \forall fo \in FO \\ t_i < fs - \Delta t_{FS} - 20ms \\ t_i > fs - \Delta t_{FS} + 20ms \end{matrix} \right\}$$

where t_i are all times at least 20ms away from all fo and fs events. To limit the computational complexity of the decoder calibration and, therefore, limit the time needed to calibrate the decoder, we selected up to 100 000 t_i by making random draws from a uniform distribution without repetitions. If less than 100 000 t_i were available, all t_i were used for calibration.

C_{FO}, C_{FS}, and C_{OTHER} were used to calibrate a multiclass regularized linear discriminant analysis (rLDA)[2] decoding model. We used previously described procedures to implement the regularization[3], using regularization parameter value of 0.5. The probability of every feature vector $F(t)$ in the test dataset to belong to the C_{FO} or C_{FS} class was then calculated using the decoder. Specifically, the decoding model was loaded into our real-time analysis application, where it was used to calculate probabilities p_{FO} and p_{FS} of the currently captured feature vector belonging to C_{FO} and C_{FS} classes. When one of these motor state probabilities crossed a threshold of 0.8, that motor state was detected. After a motor state was detected, the motor state could not be detected for the following 250ms. See **Extended Data Fig. 4** for a schematic of the decoder calibration process.

Calculation of temporal offsets

We sought to trigger stimulation over flexion and extension hotspots at the time these hotspots are active during natural locomotion. We hypothesized that these times can be decoded from neural activity by identifying neural motor states based on foot off and foot strike gait events. However, a number of temporal delays had to be taken into account to trigger the stimulation at the correct time. First, natural activation of the flexion and extension hotspots started about Δt_H=100 ms (about 10% of the gait cycle) prior to the foot off and foot strike gait events, respectively. Therefore, hotspot stimulation had to be initiated 100ms before the respective gait events. Second, we took into consideration the average latency from the control computer to the delivery of the stimulation command (See **Extended Data Fig. 2**). Third, in the majority of cases, motor state probabilities crossed the probability threshold before the motor state actually occurred, which was closer to the probability maximum.

To account for all of these latencies, we defined foot off and foot strike specific offsets, Δt_{FO} and Δt_{FS}. First, we calibrated the decoder with offsets set to 0ms and used these offsets on the data used for calibration to estimate the times when the stimulation would be initiated, e_{FO} and e_{FS}. The estimate was calculated based on the times of motor state detections modified by the median detection-to-stimulation latency. We then calculated the median difference between the estimated times of the hotspot activations and the estimated

times of the stimulations, $\Delta\hat{t}_{FO}$ and $\Delta\hat{t}_{FS}$. Times of the hotspot activations were derived from the gait event times, corrected by Δt_H.

$$\Delta\hat{t}_{FO} = M\left(fo - \Delta t_H - e_{FO}\right) \qquad \Delta\hat{t}_{FS} = M\left(fs - \Delta t_H - e_{FS}\right)$$

where M is the median operator. Only the closest estimated event to the gait event of the same type was taken into account. Median was used to exclude type I and type II errors from affecting the estimate of the offsets. We then added $\Delta\hat{t}_{RFO}$ and $\Delta\hat{t}_{RFS}$ to Δt_{FO} and Δt_{FS}, respectively and calibrated the decoder with the new temporal offsets. This procedure was repeated until $\Delta\hat{t}_{RFO}$ and $\Delta\hat{t}_{RFS}$ were between -5ms and 5ms.

We then tested the decoders over several steps of the monkey. Based on the observed kinematic effects, the Δt_H was selected, increased or decreased using 25ms increments. After completion of this calibration, Δt_{FO} and Δt_{FS} were kept constant throughout the remainder of the session.

Stimulation of hotspots triggered by detection of motor states

Upon detection of the motor states, the real-time analysis software sent a stimulation command to the Neural Research Programmer running on the control computer in order to trigger the stimulation protocols.

DATA PROCESSING AND ANALYSIS

Neuronal activity analysis

We characterized the neuronal activity by first regressing the spike event rates recorded on each electrode against the phase of the gait cycle. Thus, we obtained 96 neuronal event tuning curves, $h_{el}(\omega)$, one for each electrode *el*. Gait phase was calculated as:

$$\text{During stance:} \quad \omega(t) = 2\pi \frac{t - t_{lastFS}}{t_{nextFO} - t_{lastFS}} \cdot \bar{\omega}_{FO}$$

$$\text{During swing:} \quad \omega(t) = 2\pi \left(\bar{\omega}_{FO} + \frac{t - t_{lastFO}}{t_{nextFS} - t_{lastFO}} \left(1 - \bar{\omega}_{FO}\right) \right)$$

where t_{lastFS} and t_{lastFO} are the times of the previous foot strike and foot off events, t_{nextFS} and t_{nextFO} are the times of the next foot strike and foot off events; and $\bar{\omega}_{FO} = 0.665$ is the average proportion of the stance phase within a gait cycle over monkeys Q1-3 (Q1:0.685±0.001; Q2:0.658±0.003; Q3:0.653±0.005). The mean event rate and modulation depth for electrode *el* were calculated as the mean $h_{el}(\omega)$ over all values of ω, and the difference between the maximum and minimum of $h_{el}(\omega)$, respectively. We then fitted a double cosine function to the $h_{el}(\omega)$ in order to account for the biphasic tuning curves:

$$\hat{h}_{el}(\omega) = c + a_1 \cos\left(2\pi\left(\omega - \alpha_1\right)\right) + a_2 \cos\left(4\pi\left(\omega - \alpha_2\right)\right)$$

The preferred direction was calculated as the ω of maximum $\hat{h}_{el}(\omega)$.

The change of neuronal activity was measured by mean absolute differences of average spike event rates, modulation depths and preferred directions. These values were calculated by first calculating differences for

each electrode between two successive sessions and then calculating the mean over all electrodes. Our analysis revealed substantial neural activity changes between any two sessions, as previously observed in people with tetraplegia [4,5]. Nevertheless, the rate of change between the last pre-lesion and the first post-lesion sessions was higher than between any two other session pairs, thus suggesting ongoing adaptation of cortical activity following spinal cord lesion (see [6] for a similar finding). While the modulation depth and preferred direction changes between two post-lesion sessions returned to pre-lesion levels, changes in mean spiking event rate remained increased, suggesting that adaptions of cortical activity were still taking place during the second week after lesion.

Decoding performance quantification

Briefly, the time from the beginning to the end of each block was subdivided into foot off, foot strike and neither epochs, $T_{FO}(i)$, $T_{FS}(i)$ and $T_{NEITHER}$, defined as follows:

$$T_{FO}(i) = \left[fo(i) - \delta - \Delta t_{FO} - \tau_{TOL}, fo(i) - \delta - \Delta t_{FO} + \tau_{TOL} \right]$$
$$T_{FS}(i) = \left[fs(i) - \delta - \Delta t_{FS} - \tau_{TOL}, fs(i) - \delta - \Delta t_{FS} + \tau_{TOL} \right]$$
$$T_{NEITHER} = \left\{ t \middle| \forall i, j \;\; t \notin T_{FO}(i), t \notin T_{FS}(j) \right\}$$

where δ is the median command-to-stimulation delay; and τ_{TOL}=125ms is the detection tolerance. We counted all T_{FO} and T_{FS} epochs that contained a detected foot off or foot strike motor event as foot off and foot strike true positives (N_{TP_FO}, N_{TP_FS}), respectively. T_{FO} and T_{FS} epochs that contained no detected foot off or foot strike motor events were counted as foot off and foot strike false negatives (N_{FN_FO}, N_{FN_FS}), while the epochs that contained a motor state detection of the other kind counted as foot off / foot strike false positive and foot strike / foot off false positive ($N_{FP\_FO/FS}$, $N_{FP\_FS/FO}$). We then counted the number of decoded foot off and foot strike motor states that fell within the $T_{NEITHER}$ epoch and counted them as foot off or foot strike false positive (N_{FP_FO}, N_{FP_FS}). The number of true negative events was calculated by dividing the total length of the $T_{NEITHER}$ epoch by $2\tau_{TOL}$=250ms and then subtracting the number of foot off or foot strike false positives (N_{TP_N}). Numbers of all these events were then summed up over the trials of the same condition within a session. Confusion matrix C was calculated by normalizing the numbers of events by the respective number of epochs:

$$C = \begin{bmatrix} N_{TP\_FO}/N_{FO} & N_{FP\_FO/FS}/N_{FO} & N_{FN\_FO}/N_{FO} \\ N_{FP\_FS/FO}/N_{FS} & N_{TP\_FS}/N_{FS} & N_{FN\_FS}/N_{FS} \\ N_{FP\_FO}/N_{N} & N_{FP\_FS}/N_{N} & N_{TP\_N}/N_{N} \end{bmatrix}$$

To compare the performances of decoders, we calculated the normalized mutual information[7] C_{YX} between the actual gait events and decoded motor states as follows:

$$I(X,Y) = \sum_X \sum_Y p(x,y) \log_2 \left(\frac{p(x,y)}{p(x)p(y)} \right)$$
$$H(X) = -\sum_X p(x) \log_2 \left(p(x) \right)$$
$$C_{YX} = \frac{I(X,Y)}{H(X)}$$

where x marks all the actual gait events, y marks all the decoded motor states, $p(x)$ and $p(y)$ mark the cumulative probabilities; $p(x,y)$ mark all the joint probabilities; and X and Y are the sets of all x and y events. For a total number of actual gait events $N = N_{FO} + N_{FS} + N_N$, the probabilities are then calculated as follows:

$$p(fo,fo) = \frac{N_{TP\_FO}}{N} \qquad p(fo,fs) = \frac{N_{FP\_FO/FS}}{N} \qquad p(fo,ne) = \frac{N_{FN\_FO}}{N}$$

$$p(fs,fo) = \frac{N_{FP\_FS/FO}}{N} \qquad p(fs,fs) = \frac{N_{TP\_FS}}{N} \qquad p(fs,ne) = \frac{N_{FN\_FS}}{N}$$

$$p(ne,fo) = \frac{N_{FP\_FO}}{N} \qquad p(ne,fs) = \frac{N_{FP\_FS}}{N} \qquad p(ne,ne) = \frac{N_{TP\_N}}{N}$$

$$p(x) = \sum_{y} p(x,y) \qquad p(y) = \sum_{x} p(x,y)$$

Standard error of the C_{YX} was calculated by bootstrapping with 10 000 resamples[8].

Chance level decoding

To establish whether the accuracy of our decoders was significantly higher than chance, we designed a chance level decoder. This decoder was designed to make a decision every 20ms on whether to detect a foot off or a foot strike event with a "chance" detection probability. This probability was calibrated by taking the data used to train our real-time decoders, containing N_t time points, and decoding foot off and foot strike events using chance detection probabilities p_C that ranged from 0.08% to 0.13% in steps of 0.0005%. Detection was performed by drawing N_t values from a uniform [0,1] distribution, one for each time point. Time points with the values lower or equal to p_C were detected as foot strike events, while the values higher or equal to $1\text{-}p_C$ were detected as foot off events. Testing of each of these chance level decoders was repeated 100 times. We calculated C_{YX} for each test and selected the probability for which the chance level decoder gave the highest mean C_{YX}. In all cases, there was a clear global maximum within the tested range of probabilities, typically around 0.1%. The chance level decoder using the selected probability was then applied N_T=1000 times to the brain-controlled blocks. We measured C_{YX} for each of these tests, C_{YX}^{CL}, and calculated the p-value for significance from our actual decoder using the Monte Carlo method:

$$p = \frac{N_{GE} + 1}{N_T + 1}$$

where N_{GE} is the number of C_{YX}^{CL} higher than the C_{YX} of our real-time decoder calculated on the same set of trials.

Random decoder

To test whether our real-time decoder effectively extracted information from the neural data, we designed a "random decoder" that was calibrated using the same neural data, but randomizing the gait events in time. This allowed us to preserve all the statistical properties of the neural data. To preserve the statistical properties of gait events as much as possible, we generated randomized gait events that started with the same gait events as in the actual trial and then interchanged between the two types of gait events. We first calculated the average stance duration, τ_S, and average swing duration, τ_W, from all data that was used to build a decoder. The following iterative process was then used to generate random events, fo^* and fs^*, until there are as many random events as there are actual ones:

$$fo^*(1) = fo(1) + \frac{\tau_S}{2}U$$

$$fs^*(i) = fo^*(i) + \tau_W + \frac{\tau_W}{2}U$$

$$fo^*(i+1) = fs^*(i) + \tau_S + \frac{\tau_S}{2}U$$

where U is a number drawn from a $[-1,1]$ uniform distribution. We used these randomized events to calibrate a random decoder and applied it to the neural data collected during the brain-control blocks. We used the decoded gait events to calculate the C_{YX} of the random decoder, C_{YX}^{RAND}. This procedure was repeated N_R=1000 times, thus providing us with 1000 C_{YX}^{RAND} values. When comparing the performance of the random decoder to our real-time decoder, we used the Monte Carlo method to calculate the p value.

Spatiotemporal maps of motoneuron activation

Spatiotemporal maps of motoneuron activation were constructed by adding up the contributions of each muscle to the total activity at each spinal segment. The motor output pattern of each spinal segment Si was estimated by the following equation:

$$S_i = \frac{\sum_{j=1}^{n_i} \frac{MN_{ij}}{M_j} EMG_j}{\sum_{j=1}^{n_i} \frac{MN_{ij}}{M_j}} \qquad M_j = \sum_{k=1}^{n_j} MN_{kj}$$

where EMG_j represents the envelope of muscle activity normalized to the maximum value observed across all experimental sessions for each muscle, n_i is the number of EMG_js corresponding to the i-th segment, MN_{ij} is the number of motor neurons of j-th muscle in the i-th segment, and M_j is the total number of motor neurons of the j-th muscle (**Extended Data Fig. 3a**).

SUPPLEMENTARY TABLES

Animal ID	Micro electrode array	Spinal implant	Wireless muscle activity system	Spinal cord injury	Retrograde motoneuron tracing	Intact walking experiments	Single pulse spinal cord stimulation	Pre-lesion brain-control stimulation	Post-lesion brain-control stimulation
Q1	✓	✓	✓	-	✓	✓	✓	✓	-
Q2	✓	✓	✓	✓	✓	✓	✓	✓	✓
Q3	✓	✓	*	✓	✓	✓	*	*	✓
P1	✓	✓	✓	-	-	-	✓	-	-
P2	✓	-	✓	-	✓	✓	-	-	-
P3	✓	-	✓	-	-	✓	-	-	-
P4	-	-	-	-	✓	-	-	-	-
P5	-	-	-	-	✓	-	-	-	-
P6	-	-	-	-	✓	-	-	-	-

✓ Performed
- Not Performed
* Could not be performed due to technical issues or animal conditions

Supplementary Table 1. Experimental procedures conducted on the animals.

Monkey	Session	# of blocks Stimulation Off	# of blocks Brain control: Flexion	# of blocks Brain control: Extension	# of blocks Brain Control: Both	# of blocks Continuous stimulation
Q1	Pre-lesion 1: treadmill 28.7.2014	**5**		1 at 40 Hz	4 at 40 Hz	
	Pre-lesion 2: treadmill 29.7.2014	**4**	1 at 40Hz	1 at 40 Hz	4 at 40 Hz	
	Pre-lesion 3: treadmill 30.7.2014.	**4**	5 at 40 Hz		5 at 40 Hz	
Q2	Pre-lesion 1: treadmill 24.3.2015.	**2**	**3 at 60 Hz** 1 at 60 Hz	**3 at 60 Hz** 1 at 60 Hz	3 at 60 Hz	
	Pre-lesion 2: treadmill 26.3.2015.	**4**	**2 at 60 Hz** 1 at 60 Hz	**2 at 60 Hz**	3 at 60Hz	
	Pre-lesion 3: treadmill 28.3.2015.	**3**	**2 at 60 Hz** 4 at 30 to 80 Hz	**2 at 60 Hz**		
	Post-lesion day 6: treadmill 23.06.2015.	**4**			4 at 40Hz	
	Post-lesion day 7: corridor 24.06.2015.	**5** 2	**2 at 40 Hz**	**3 at 40 Hz**	2 at 40Hz	
	Post-lesion day 13: treadmill 30.06.2015.	**6**			13 at 20 to 60 Hz	1 at 40Hz
	Post-lesion day 14: Corridor 01.07.2015.	**6** 1			6 at 30Hz	
	Post-lesion day 99: treadmill 24.9.2015.	8				
Q3	Pre-lesion 1: treadmill 16.1.2015.	2				
	Post-lesion day 16: treadmill 3.7.2015.	**2**	**2 at 40Hz**	**2 at 60Hz**	8 at 30 to 80Hz	
	Post-lesion day 99: treadmill 24.9.2015.	8				
P2	Intact: treadmill 26.11.2013.	1				
P3	Intact : treadmill 12.12.2013.	1				

Supplementary table 2. Behavioral experiments conducted with each animal. Trials in bold and underlined were used to calibrate the brain decoders used on that session.

PARAMETERS	VARIABLE	DETAILED EXPLANATION
Limb endpoint (Metatarsal phalange) trajectory		
	1	Step length
	2	Maximum backward position of the foot
	3	Maximum forward position of the foot
	4	Step height
	5	Speed of the foot at swing onset
	6	Angular acceleration of the foot velocity vector at swing onset
Stability		
Trunk and pelvic position and oscillations	7	Maximum hip sagittal position
	8	Minimum hip sagittal position
	9	Length of pelvis displacements in the forward direction
	10	Length of pelvis displacements in the vertical direction
Joint angles		
Backward	11	Crest oscillations
	12	Thigh oscillations
	13	Leg oscillations
	14	Foot oscillations
Forward	15	Crest oscillations
	16	Thigh oscillations
	17	Leg oscillations
	18	Foot oscillations
Flexion	19	Knee joint angle
	20	Ankle joint angle
Abduction	21	Whole limb abduction
	22	Foot abduction
Extension	23	Knee joint angle
	24	Ankle joint angle
Adduction	25	Whole limb adduction
	26	Foot adduction

Supplementary table 3 | Computed kinematic parameters.

REFERENCES

1 Wenger, N. *et al.* Spatiotemporal neuromodulation therapies engaging muscle synergies improve motor control after spinal cord injury. *Nature medicine* **22**, 138-145, doi:10.1038/nm.4025 (2016).

2 Friedman, J. Regularized Discriminant-Analysis. *J Am Stat Assoc*, 165-175 (1989).

3 Milekovic, T., Ball, T., Schulze-Bonhage, A., Aertsen, A. & Mehring, C. Detection of error related neuronal responses recorded by electrocorticography in humans during continuous movements. *PLoS One* **8**, e55235, doi:10.1371/journal.pone.0055235 (2013).

4 Jarosiewicz, B. *et al.* Virtual typing by people with tetraplegia using a self-calibrating intracortical brain-computer interface. *Science translational medicine* **7**, 313ra179, doi:10.1126/scitranslmed.aac7328 (2015).

5 Perge, J. A. *et al.* Intra-day signal instabilities affect decoding performance in an intracortical neural interface system. *J Neural Eng* **10**, 036004, doi:10.1088/1741-2560/10/3/036004 (2013).
6 Hollis, E. R., 2nd *et al.* Ryk controls remapping of motor cortex during functional recovery after spinal cord injury. *Nat Neurosci* **19**, 697-705, doi:10.1038/nn.4282 (2016).
7 MacKay, D. J. C. *Information theory, inference, and learning algorithms*. (Cambridge University Press, 2003).
8 Moore, D. S., McCabe, G. P. & Craig, B. *Introduction to the practice of statistics*. 6th ed. edn, (W.H.Freeman, 2009).