University of South Australia

UniSA STEM

Mathematics Clinic 2024

Better energy estimates for train journeys

Final Report for

Trapeze Group

November 2024

Students

Liam Connors

Huyen Thi Thu Pham

Sophie Vince

Shuo Zhang

Academic Advisor

Peter Pudney

Academic Consultant

Peng Zhou

Clinic Director

Peter Pudney

Liaisons

Farbod Nejati

Trapeze™

Here for the journey

Abstract

Trapeze's Energymiser Driving Advice System uses GPS technology to monitor a train journey, then formulates an optimal driving strategy to ensure punctuality while minimising energy consumption. Energymiser estimates energy consumption from observed speed profiles, but these estimates have poor accuracy. Data logs from locomotives provide additional data on how the driver controlled each train, and this data can be used to provide better estimates of energy use. These logs also provide valuable insights into how drivers respond to driving advice throughout the journey.

This report presents findings of our Mathematics Clinic project to find better ways of estimating energy use. We show that energy calculated by Energymiser does not always match energy calculated from time-in-notch, we explain and demonstrate how inaccurate speed measurements can lead to overestimates in energy consumption, and we show how sampled speed observations or an Unscented Kalman Filter could be applied to provide better estimates of train speed. We also compare the actual control of trains to Energymiser advice as a way to get a better view of how effectively the Energymiser system is being used, and demonstrate a better way of comparing energy consumption of freight journeys.

Acknowledgements

We would like to thank Trapeze and our industry liaison Farbod Nejati for the opportunity to work on this project, Academic Advisor and Clinic Director Peter Pudney for organising the project and offering his expertise to our project. We thank our Academic Consultant Peng Zhou for his hard work and expertise. We also thank Lesley Ward and Alex Tam for their assistance with the reports.

All student team members participated, and held leadership roles, in the team.

Team Member	Role Semester 1	Role Semester 2
Liam Connors	Editor: Report	Editor: Software
Huyen Thi Thu Pham	Project Manager	Project Manager
Sophie Vince	Editor: Work Statement	Editor: Report
Shuo Zhang	Editor: Presentation	Editor: Presentation

Contents

List of Figures

List of Tables

1 Introduction

Trapeze is an Australian company that develops technologies to assist public transport efficiency. One of their products is the Energymiser Driving Advice System, which they have developed in collaboration with researchers at the University of South Australia. Energymiser provides driving advice to the driver, in order to minimise energy usage whilst also keeping the train on time. Energymiser is used by 20 rail operators around the world, one of which is KiwiRail, a New Zealand rail operator. Trapeze have been able to access data logs from a number of KiwiRail journeys, providing us with data from Energymiser and data from the locomotives.

The Energymiser driving advice system uses GPS to monitor the location and speed of the train throughout a journey, and then uses train, route and timetable data to create and present a driving strategy for drivers to follow. This driving strategy is designed to minimise the amount of energy used by the train, whilst meeting all timing requirements.

Energymiser is able to calculate optimal driving strategies without measuring or estimating energy use, but rail operators are interested in knowing how much energy their trains are using. Energymiser is not able to access energy use data from the train, and therefore must estimate energy use from the observed movement of the train.

Figure 1.1 shows an example of an Energymiser screen during a journey. The lower half of the screen shows the track gradient profile and trackside features. The short white line indicates the location of the train (this particular train was quite short. The top half of the display shows the track speed limits in orange and the advised speed profile as the thick line. The colour of the speed profile indicates the advised driving mode: green for power, and white for coast. In Figure 1.1, Energymiser is advising the driver to travel at the speed limit of 70 km/h, and then coast. The Energymiser system is an advice system, and it is the driver's responsibility to control the train.

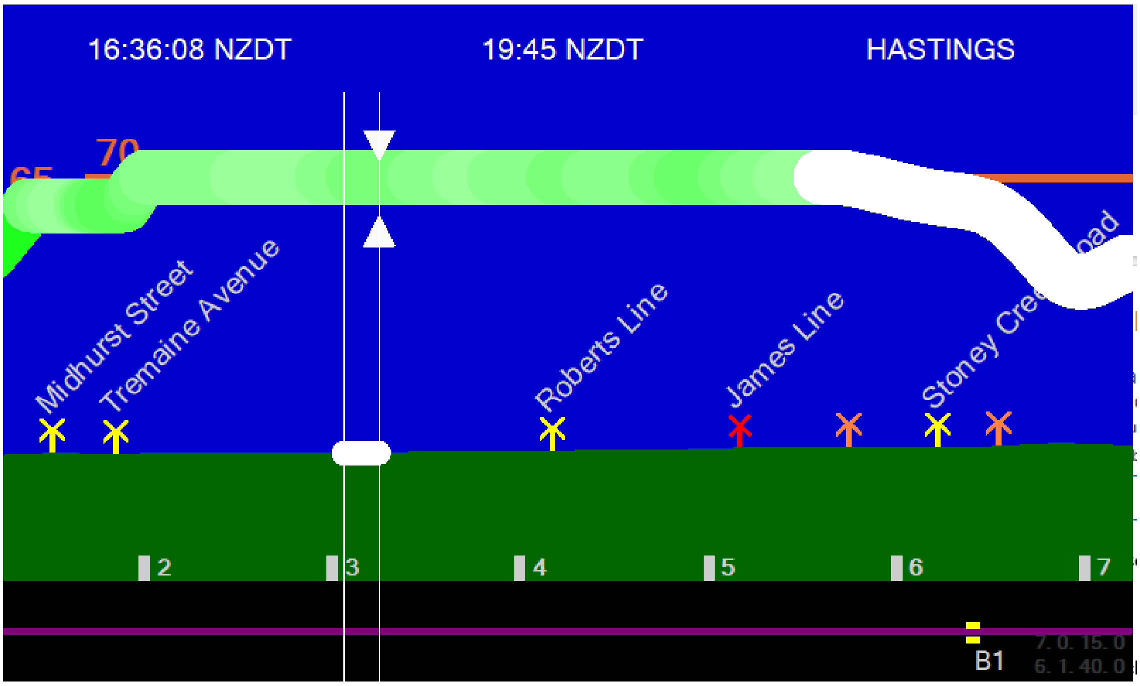


Figure 1.1: Screenshot of Energymiser display during a KiwiRail journey.

In this report we begin by looking at the Energymiser and locomotive data provided to us by Trapeze for KiwiRail journeys (Section 2). Section 3 summarises how Energymiser calculates energy, and shows how speed inaccuracies can lead to overestimates of energy use.

Section 3 also shows how energy can be calculated from notch data provided in the locomotive logs, and how improved estimates of energy use can be obtained by sampling speed at wider intervals.

Section 4 shows how mathematical filters can be used to smooth speed data, and how an Unscented Kalman filter can be used to estimate speed and energy.

In Section 5 we compare how the driver controlled a train to the advice that was given.

Section 6 develops regression models for estimating the energy consumption for trains with different lengths, masses and speeds, and shows how these can be used to identify journeys that use more energy than expected.

Section 7 gives a summary of our findings. Also included is a reference list to the sources that assisted our research.

2 Data

Trapeze have provided us with pairs of data logs from about 50 KiwiRail train journeys or journey segments, run between July 2023 and March 2024. Each pair of logs contains an Energymiser log and a log from the locomotive.

In this part of the report:

- Section 2.1 explains how we combined the data from the Energymiser log and the locomotive log for each journey
- Section 2.2 shows example speed profiles from Energymiser and locomotive logs
- Section 2.3 shows results from an unsuccessful attempt to estimate speed from change in distance
- Section 2.4 shows the power and fuel flow rates for each notch of a DL locomotive.

2.1 Combined data logs

The original logs have complicated formatting and contain a lot of data that is not necessary for our analysis. Our academic advisors had software for reading these logs, so wrote software to align the two data sets based on GPS timestamps to output a single combined log for each journey. Figure 2.1 shows part of an example combined log. The shaded columns are from Energymiser, and the unshaded columns from the locomotive log.

The columns are described below:

Time is the date and time of each data record, at 1-second intervals.

Distance is the distance along the route in kilometres. Distance is calculated by Energymiser by projecting longitude and latitude from GPS onto the known path of the route.

Time	Distance (km)	Energymiser speed (km/h)	Advice	Advised speed (km/h)	Energy (J)	Longitude	Latitude	Notch	Air brake	Dynamic brake	Loco speed (km/h)
14/03/2024 2:05	169.6	77	0.544	80	539733273	175.265954	-37.815922	5	554	0	77
14/03/2024 2:05	169.611	77	0.545	80	540462220	175.265992	-37.816114	5	555	0	77
14/03/2024 2:05	169.632	77	0.546	80	541780608	175.266031	-37.816305	5	554	0	77
14/03/2024 2:05	169.652	77	0.547	80	542961292	175.266071	-37.816496	5	554	0	77
14/03/2024 2:05	169.675	78	0.548	80	544407817	175.266114	-37.816687	5	555	0	77
14/03/2024 2:05	169.708	78	0.549	80	546250822	175.266158	-37.816878	5	555	0	77
14/03/2024 2:05	169.729	78	0.549	80	547558043	175.266204	-37.817069	5	555	0	77
14/03/2024 2:05	169.751	78	0.549	80	548828024	175.26625	-37.81726	5	555	0	77
14/03/2024 2:05	169.773	78	0.548	80	550080238	175.266299	-37.81745	5	555	0	77
14/03/2024 2:05	169.794	78	0.517	80	551373066	175.266349	-37.817641	5	555	0	77
14/03/2024 2:05	169.805	78	0.501	80	552108979	175.266401	-37.817831	5	555	0	77
14/03/2024 2:05	169.827	78	0.469	80	553371393	175.266454	-37.818022	5	555	0	77
14/03/2024 2:05	169.848	78	0.437	80	554740064	175.266508	-37.818212	5	555	0	77
14/03/2024 2:05	169.881	78	0.372	80	556696578	175.266564	-37.818402	5	555	0	77
14/03/2024 2:05	169.892	78	0.355	80	557461675	175.266621	-37.818592	5	555	0	77
14/03/2024 2:05	169.911	78	0.322	80	558667944	175.26668	-37.818782	5	555	0	77
14/03/2024 2:05	169.946	78	0.272	80	560706947	175.26674	-37.818971	5	554	0	77
14/03/2024 2:05	169.968	78	0.238	80	562024316	175.266802	-37.81916	5	555	0	77
14/03/2024 2:05	169.978	78	0.221	80	562719130	175.266865	-37.819349	5	554	0	77
14/03/2024 2:05	170.011	78	0.153	80	564633974	175.26693	-37.819538	5	555	0	77
14/03/2024 2:05	170.02	78	0.153	80	565250286	175.266996	-37.819726	5	555	0	77
14/03/2024 2:05	170.055	78	0.087	80	567320793	175.267065	-37.819914	4	555	0	77
14/03/2024 2:05	170.076	78	0.054	80	568713817	175.267133	-37.820102	3	555	0	77
14/03/2024 2:05	170.098	78	0.021	80	570096476	175.267201	-37.820291	2	555	0	77
14/03/2024 2:05	170.12	78	0	80	571287342	175.267269	-37.820479	2	554	0	77
14/03/2024 2:05	170.141	78	0	80	572218398	175.267336	-37.820667	2	554	0	77
14/03/2024 2:05	170.163	78	0	79	573514064	175.267404	-37.820856	1	555	0	77
14/03/2024 2:05	170.185	78	0	79	574767103	175.267472	-37.821044	1	555	0	77
14/03/2024 2:05	170.196	78	0	79	575335019	175.267539	-37.821232	1	554	0	77

Figure 2.1: Example combined data log.

Energymiser speed is an estimate of the train speed in kilometres per hour, derived from GPS measurements. Energymiser speed will be more accurate than the locomotive speed when the GPS receiver has a clear view of the sky, but is not accurate when the sky is obscured.

Advice is a value between -1 and 1 that indicates how the train should be driven—see Table 2.1. In fact, the advice value in the log is a smoothed version of the driving advice value, and is the mean of the actual driving advice value over the next 15 seconds.

Advised speed, in km/h, is the ideal speed for the train calculated by Energymiser.

Energy is the Energymiser estimate of the cumulative work done by the traction system during the journey, in joules. Energymiser is not able to directly access how much energy has been used, so estimates energy based on the speed profile of the train.

Longitude and latitude are the position of the train, from a GPS unit in the front cab.

Notch is the position of the driver's control lever that controls the traction power being applied at the locomotive wheels. Notch setting ranges from 0 (no power), to 8 (full power).

Air brake pressure indicates the pressure in the air brake system, which decreases when the air brakes are applied. Air brakes operate on every wheel along the train, but are slow to apply and release. Dynamic brakes (next item) respond faster but are less effective. Drivers typically use dynamic brakes at high speeds and blend with air brakes at lower speeds.

Dynamic brake value is a value between 0 (off) and 1 (full dynamic braking) that indicates how much electric braking is being applied to the locomotive wheels. Dynamic braking responds faster than air brakes, but applies braking force to the locomotive wheels only.

Locomotive speed is the speed the locomotive in km/h. It is derived from the rotation of the locomotive wheels. Accuracy varies with the diameter of the wheels—the apparent speed of the train increases as the wheels wear.

Table 2.1: Advice values and driving modes.

Advice value	Driving mode	
$a = 1$	Power	Apply maximum driving power
$0 < a < 1$	Hold	Maintain constant speed
$a = 0$	Coast	No power, no braking
$a = -1$	Brake	Apply maximum braking

Table 2.2 lists the journeys used in our analysis. Most logged journeys had trains hauled by 1 DL locomotive (2208 kW) or by 2 DL locomotives (4417 kW), as shown in Table 2.2, but for some logs the number of locomotives being used changed during the journey. There are also trips where the train length and mass change during the journey. We divided journeys where the length, mass or locomotive power change during the journey into separate journeys.

Table 2.2: Train data.

Journey	Length (m)	Mass (tonnes)	Power (kW)
20230720_001308_212_KR-DL9008B	571	1156	2208
20230720_004420_620_KR-DF7173	469	962	3167
20230720_062403_924_KR-DF7132	153	516	1583
20240130_073151_140_KR-DL9291B	280	1218	2208
20240130_101838_141_KR-DL9291B	280	453	2208
20240130_220634_144_KR-DL9291B	295	1284	2208
20240201_081944_MP6_KR-DL9291A	735	1696	2208
20240201_193532_MP5_KR-DL9291B	744	1858	2208
20240209_114205_MP3_KR-DL9291B	740	1908	2208
20240209_191900_MP10_KR-DL9291A	669	1278	2208
20240210_054351_MP1_KR-DL9291B	737	1830	2208
20240210_161266_MP20_KR-DL9291A	737	1310	2208
20240213_185702_521_KR-DL9291B	574	1399	4417
20240214_125404_535_KR-DL9291B	305	1186	2208, 4417
20240219_200352_215_KR-DL9291B	351	1154	2208, 4417
20240220_065630_670_KR-DL9291A	100	208	2208
20240220_111847_671_KR-DL9291B	203	918	2208
20240220_121837_671_KR-DL9291A	203	918	2208
20240221_113610_390_KR-DL9291A	389	989	2208
20240221_182358_391_KR-DL9291B	387	652	2208
20240221_231012_323_KR-DL9291A	700	1808	2208
20240222_020159_328_KR-DL9291A	500	808	2208
20240223_014666_566_KR-DL9291A	511	759	2208
20240223_124351_565_KR-DL9291B	511	1173	2208
20240223_220737_265_KR-DL9291B	466	1027	2208
20240224_035366_234_KR-DL9291A	649	1661	2208, 4417
20240226_015821_MP9_KR-DL9291B	720	1498	2208
20240226_115932_MP8_KR-DL9291A	720	1692	2208
20240227_003741_183_KR-DL9291B	411	546	2208
20240227_081322_321_KR-DL9291B	689	1966	2208
20240227_115742_MP8_KR-DL9291A	740	1688	2208
20240228_170466_212_KR-DL9291A	497	1373	4417
20240313_203920_541_KR-DL9475A	260	796	2208, 4417
20240314_001159_228_KR-DL9538A	406	1118	2208, 4417
20240314_014931_391_KR-DL9072B	555	655	2208
20240314_100234_352M_KR-DL9688B	437	1488	2208

2.2 Speed profiles

Figure 2.2 shows Energymiser speed (blue) and locomotive speed (orange) from an example log. There are several locations where there is a significant difference between the two speeds. Looking at the longitude and latitude of the train and using Google Earth, we can show that drops in Energymiser speed often correspond to the train entering a tunnel (an example is shown in Figure 2.3).

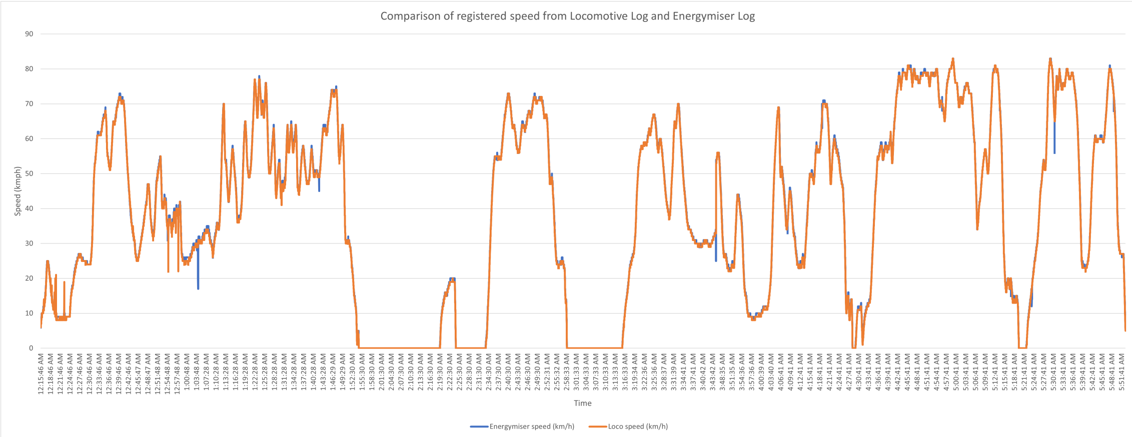


Figure 2.2: Speed profiles from Energymiser (blue) and the locomotive log (orange).

We can compare the speed profile with the notch setting profile (Figure 2.4) in order to analyse how the train is being driven. Figure 2.5 shows a closer view of a journey section where the driver is applying full power but the speed is increasing and decreasing because of steep track gradients.

Figure 2.3: Google Earth image showing the position of a train going into a tunnel.

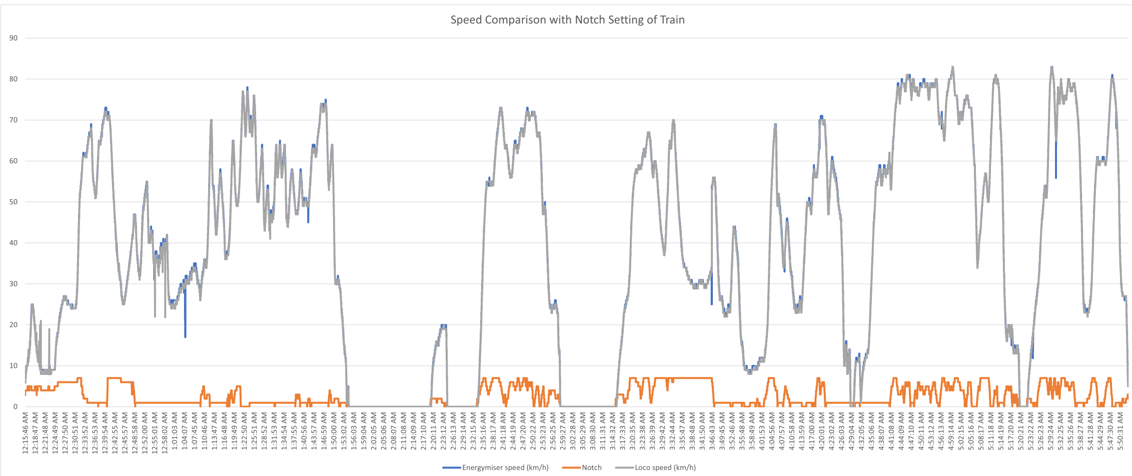


Figure 2.4: Speed profile and notch setting.

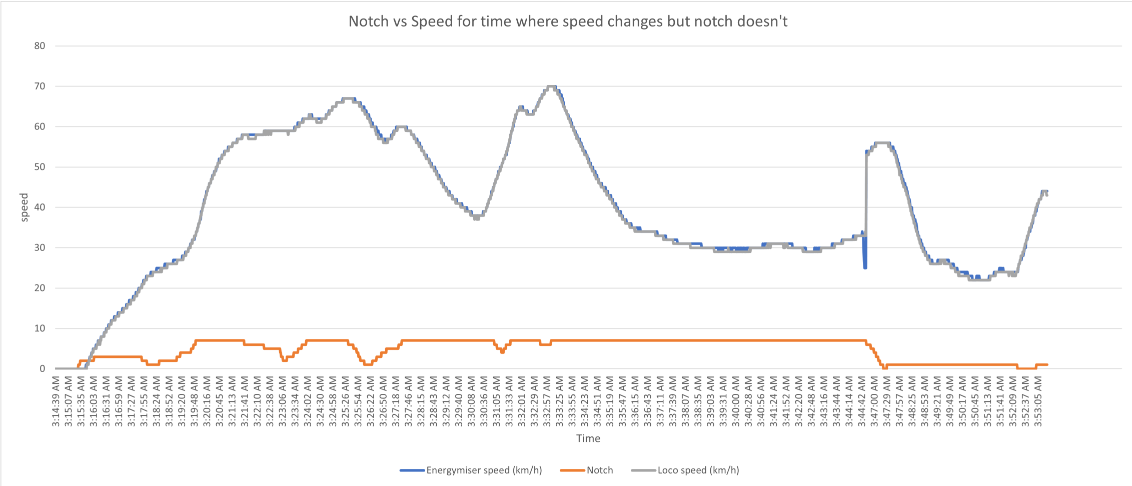


Figure 2.5: Speed and notch with steep gradients.

2.3 Speed from distance

We will show in Section 3 that noisy speed observations cause energy to be over-estimated. We investigated whether we could estimate speed from the change in observed distance over time. Figure 2.6 shows the same speed profiles as in Figure 2.2. The grey lines indicate the speed estimated by

$$\bar{v}_i = \frac{x_{i+1} - x_i}{t_{i+1} - t_i}$$

where x_i are speed observations and t_i are time observations.

As seen in Figure 2.6, estimates of speed from distance observations are worse than the GPS speed. This is because distance observations are calculated by Energymiser by projecting noisy GPS positions onto the known path of the track.

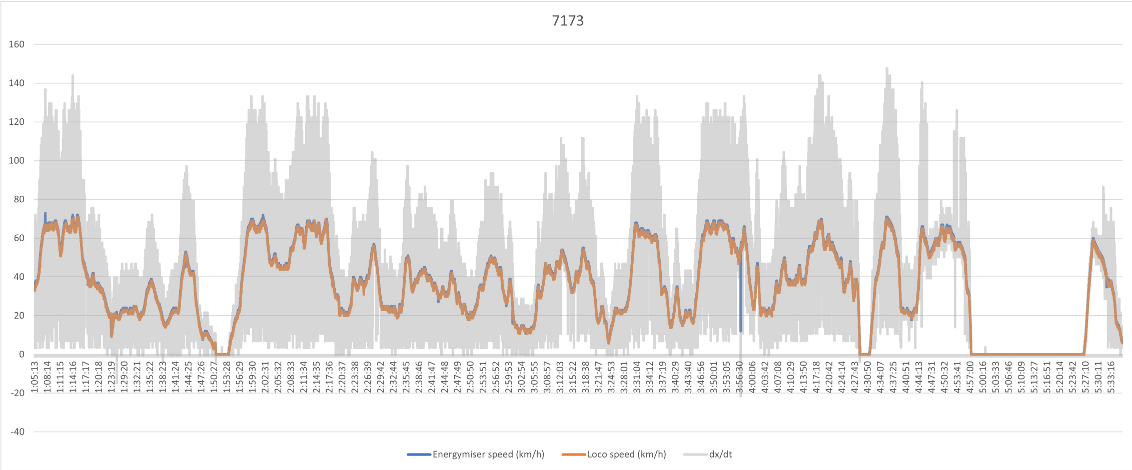


Figure 2.6: Speed profile from Energymiser and locomotive logs, with a calculated speed estimate from distance data.

2.4 Power in notch

Table 2.3 shows the power generated at the locomotive wheels and fuel flow rates for each notch of a DL locomotive. This table is based on data provided by KiwiRail, which provided generator power. We have shifted and scaled the generator power to calculate power at the wheel that corresponds with the data in the tractive effort tables of the Energymiser train files.

Table 2.3: Power and fuel flow rates of a DL locomotive.

Notch	Power (kW)	Fuel rate (L/h)
0	0	10
1	25	36
2	217	70
3	380	120
4	615	190
5	990	310
6	1393	410
7	1939	550
8	2208	680

3 Estimating energy from speed and time-in-notch

This section of the report shows how Energymiser estimates the energy use of a train from speed observations (Section 3.1), and uses a simple simulation to show how noisy speed observations can lead to overestimation of energy use (Section 3.1). We then show how to calculate energy from the time spent in each control notch (Section 3.3), and how sampling speed at 10-second intervals can give improved energy estimates (Section 3.5).

3.1 Energymiser energy calculation

As the Energymiser system does not have access to energy consumption information, it estimates energy use from the observed speed of the train. We used a simple spreadsheet to demonstrate the calculation and what can go wrong. Figure 3.1 shows our spreadsheet.

Train parameters were taken from an example Energymiser log:

- train mass $m = 1156\,\text{tonnes}$
- resistance coefficients $r_0 = 15\,767\,\text{N}$, $r_1 = 309.18\,\text{N/(m/s)}$, $r_2 = 29.59\,\text{N/(m/s)}^2$.

We assumed that:

- the train is travelling on a flat track, so we can ignore gradient forces
- the train starts at speed $v = 10\,\text{m/s}$
- the driver applies 2200 kW of power for the first 30 seconds then coasts for the next 30 seconds
- the state of the train is calculated at time steps of $\Delta t = 1\,\text{s}$.

The motion of the train is approximated by the following equations:

- distance: $d_{i+1} = d_i + v_i \Delta t + \frac{1}{2} a_i \Delta t^2$

Parameter	Value	Units	Time (s)	Distance (m)	Speed (m/s)	Applied power (kW)	Tractive force (kN)	Resistance (kN)	Acceleration (m/s2)	Actual energy (kJ)	Accumulated control energy (kJ)
mass	1156	tonnes	0	0.0	10.00	2200	220.00	21.82	0.171	2200	0
resistance r0	15767	N	1	10.1	10.17	2200	216.29	21.97	0.168	2200	2200
resistance r1	309.18	N/(m/s)	2	20.3	10.34	2200	212.78	22.13	0.165	2200	4400
resistance r2	29.59	N/(m/s)^2	3	30.8	10.50	2200	209.43	22.28	0.162	2200	6600
			4	41.3	10.67	2200	206.26	22.43	0.159	2200	8800
time step	1	s	5	52.1	10.83	2200	203.23	22.58	0.156	2200	11000
			6	63.0	10.98	2200	200.33	22.73	0.154	2200	13200
			7	74.1	11.14	2200	197.57	22.88	0.151	2200	15400
			8	85.3	11.29	2200	194.93	23.03	0.149	2200	17600
			9	96.6	11.44	2200	192.39	23.17	0.146	2200	19800
			10	108.1	11.58	2200	189.96	23.32	0.144	2200	22000
			11	119.8	11.73	2200	187.62	23.46	0.142	2200	24200
			12	131.6	11.87	2200	185.38	23.60	0.140	2200	26400
			13	143.5	12.01	2200	183.22	23.75	0.138	2200	28600
			14	155.6	12.15	2200	181.14	23.89	0.136	2200	30800
			15	167.8	12.28	2200	179.13	24.03	0.134	2200	33000
			16	180.2	12.42	2200	177.19	24.17	0.132	2200	35200
			17	192.6	12.55	2200	175.33	24.31	0.131	2200	37400
			18	205.3	12.68	2200	173.52	24.44	0.129	2200	39600
			19	218.0	12.81	2200	171.77	24.58	0.127	2200	41800
			20	230.9	12.94	2200	170.08	24.72	0.126	2200	44000
			21	243.9	13.06	2200	168.44	24.85	0.124	2200	46200
			22	257.0	13.18	2200	166.86	24.99	0.123	2200	48400
			23	270.2	13.31	2200	165.32	25.12	0.121	2200	50600
			24	283.6	13.43	2200	163.82	25.26	0.120	2200	52800
			25	297.1	13.55	2200	162.38	25.39	0.119	2200	55000
			26	310.7	13.67	2200	160.97	25.52	0.117	2200	57200
			27	324.4	13.78	2200	159.60	25.65	0.116	2200	59400
			28	338.3	13.90	2200	158.27	25.78	0.115	2200	61600
			29	352.2	14.02	2200	156.97	25.91	0.113	2200	63800
			30	366.3	14.13	0	0.00	26.04	-0.023	0	66000
			31	380.4	14.11	0	0.00	26.02	-0.023	0	66000
			32	394.5	14.08	0	0.00	25.99	-0.022	0	66000
			33	408.6	14.06	0	0.00	25.96	-0.022	0	66000
			34	422.6	14.04	0	0.00	25.94	-0.022	0	66000
			35	436.7	14.02	0	0.00	25.91	-0.022	0	66000
			36	450.7	13.99	0	0.00	25.89	-0.022	0	66000
			37	464.7	13.97	0	0.00	25.86	-0.022	0	66000
			38	478.6	13.95	0	0.00	25.84	-0.022	0	66000
			39	492.5	13.93	0	0.00	25.81	-0.022	0	66000
			40	506.5	13.90	0	0.00	25.79	-0.022	0	66000

Figure 3.1: Spreadsheet for demonstrating energy calculation.

- speed: $v_{i+1} = v_i + a_i \Delta t$
- tractive force: $F = P/v$
- resistance: $R = r_0 + r_1 v + r_2 v^2$
- acceleration: $a = \dfrac{F - R}{m}$
- energy: $P \Delta t$

Figure 3.2 shows the speed profile of our simulated train journey. Train speed increases for the first 30 seconds, when there is power being applied. For the remaining 30 seconds, when no power is being applied, the train speed gradually decreases.

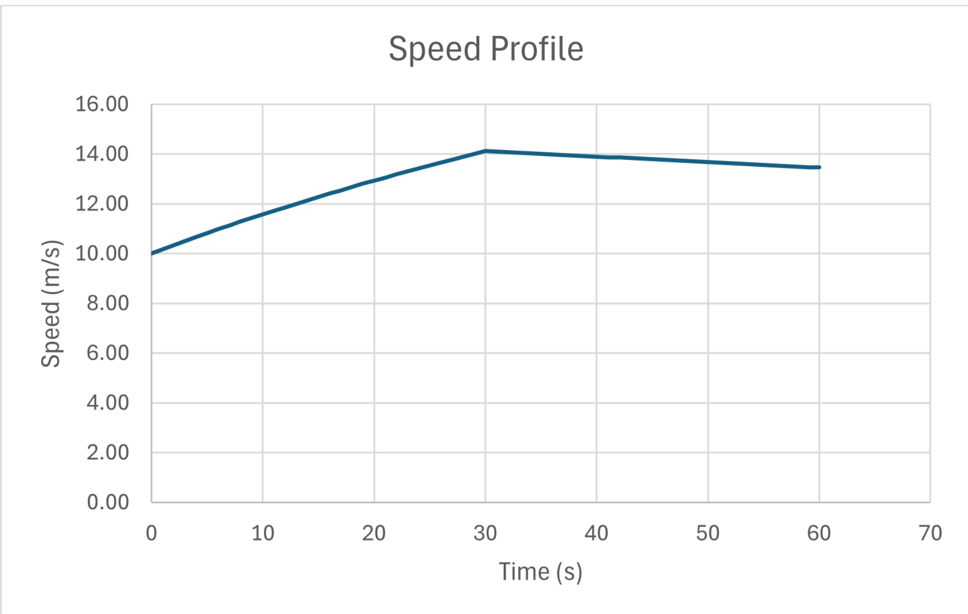


Figure 3.2: Simulated speed profile.

We used our spreadsheet, shown in Figure 3.3, to calculate energy from speed observations, using the same method used by Energymiser.

Observed speed (m/s)	ΔEk	Resistance work	Estimated force (kN)	Tractive force (kN)	Tractive energy (kJ)	Accumulated observed energy (kJ)
10.00	1998.81	220.84	440.16	440.16	112.1	0
10.17	1992.83	226.14	216.37	216.37	2219.0	112.1
10.34	1986.94	231.41	212.85	212.85	2218.3	2331.1
10.50	1981.11	236.65	209.51	209.51	2217.8	4549.4
10.67	1975.36	241.86	206.33	206.33	2217.2	6767.2
10.83	1969.66	247.04	203.30	203.30	2216.7	8984.4
10.98	1964.02	252.19	200.41	200.41	2216.2	11201.1
11.14	1958.44	257.32	197.64	197.64	2215.8	13417.3
11.29	1952.90	262.42	195.00	195.00	2215.3	15633.1
11.44	1947.41	267.51	192.46	192.46	2214.9	17848.4
11.58	1941.97	272.57	190.03	190.03	2214.5	20063.3
11.73	1936.57	277.61	187.70	187.70	2214.2	22277.9
11.87	1931.20	282.63	185.45	185.45	2213.8	24492.0
12.01	1925.87	287.63	183.29	183.29	2213.5	26705.9
12.15	1920.57	292.61	181.21	181.21	2213.2	28919.4
12.28	1915.31	297.57	179.20	179.20	2212.9	31132.5
12.42	1910.08	302.52	177.26	177.26	2212.6	33345.4
12.55	1904.87	307.45	175.39	175.39	2212.3	35558.0
12.68	1899.70	312.37	173.59	173.59	2212.1	37770.4
12.81	1894.55	317.27	171.84	171.84	2211.8	39982.4
12.94	1889.42	322.16	170.15	170.15	2211.6	42194.2
13.06	1884.32	327.03	168.51	168.51	2211.4	44405.8
13.18	1879.24	331.89	166.92	166.92	2211.1	46617.2
13.31	1874.19	336.73	165.38	165.38	2210.9	48828.3
13.43	1869.15	341.56	163.89	163.89	2210.7	51039.2
13.55	1864.14	346.38	162.44	162.44	2210.5	53249.9
13.67	1859.14	351.19	161.03	161.03	2210.3	55460.5
13.78	1854.17	355.99	159.66	159.66	2210.2	57670.8
13.90	1849.21	360.77	158.33	158.33	2210.0	59881.0
14.02	1844.27	365.54	157.04	157.04	2209.8	62090.9
14.13	-367.63	367.45	-0.01	0.00	0.0	64300.7
14.11	-366.68	366.50	-0.01	0.00	0.0	64300.7
14.08	-365.74	365.56	-0.01	0.00	0.0	64300.7
14.06	-364.79	364.61	-0.01	0.00	0.0	64300.7
14.04	-363.85	363.67	-0.01	0.00	0.0	64300.7
14.02	-362.91	362.73	-0.01	0.00	0.0	64300.7
13.99	-361.97	361.80	-0.01	0.00	0.0	64300.7
13.97	-361.04	360.86	-0.01	0.00	0.0	64300.7
13.95	-360.11	359.93	-0.01	0.00	0.0	64300.7
13.93	-359.18	359.00	-0.01	0.00	0.0	64300.7
13.90	-358.25	358.07	-0.01	0.00	0.0	64300.7

Figure 3.3: Calculating energy from speed observations.

The equations used to calculate the work done by the traction system during any interval $[t_0, t_1]$ are:

- change in kinetic energy: $\Delta E_{\mathrm{k}} = \frac{1}{2}m(v_1^2 - v_0^2)$
- resistance work: $W_r = \dfrac{v_0 R(v_0) + v_1 R(v_1)}{2}\Delta t$
- estimated force: $F = \dfrac{\Delta E_k + W_r}{(v_0 + v_1)\Delta t/2}$
- tractive force $= F^+$
- tractive energy $= F^+(v_0 + v_1)\Delta t/2$.

In the next two sections we investigate what happens if there are inaccuracies in these speed measurements, such as spikes or rounding.

3.2 Causes of overestimation

Overestimation of energy use can occur because of spikes in the GPS observations, or from inaccuracies in the speed measurements due to rounding.

3.2.1 Speed spikes

Upward or downward spikes in observed speed occur when speed observations are higher or lower than the actual speed for a short duration. After comparing speed profiles to train route data, we found spikes often occur as a train enters a tunnel.

For our example, we change an actual speed of 11.44 m/s to 12 m/s for one sample, as shown in Table 3.1.

Table 3.1: Actual and observed speeds for a spike.

Actual (m/s)	Observed (m/s)
10.83	10.83
10.98	10.98
11.14	11.14
11.29	11.29
11.44	12.00
11.58	11.58
11.73	11.72
11.87	11.87
12.01	12.01

Figure 3.4 shows the resulting change in the estimated energy profile. The blue curve is the actual energy use and the orange curve is the estimated energy use. The spike in speed causes energy to be overestimated, and the error does not get corrected when the observed speed returns to its correct value.

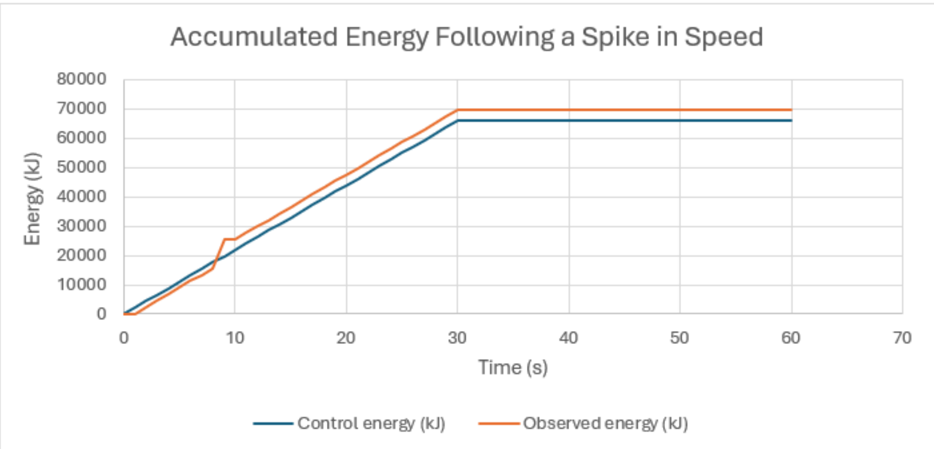


Figure 3.4: Change in estimated energy due to a speed spike.

3.2.2 Rounded speed observations

Overestimates of energy can also be caused by small errors in the speed observations. We show what happens to the energy estimate if the observed speeds are rounded to the nearest km/h, as GPS does. Table 3.2 shows the actual speeds from the simulation and the rounded speeds (converted from rounded km/h back to m/s).

Table 3.2: Actual and observed speeds for rounded speeds.

Actual (m/s)	Observed (m/s)
10.00	10.00
10.17	10.28
10.34	10.28
10.50	10.56
10.67	10.56
10.83	10.83
10.98	11.11
11.14	11.11
11.29	11.39
11.44	11.39

Figure 3.5 shows the effects of rounding on the energy calculation.

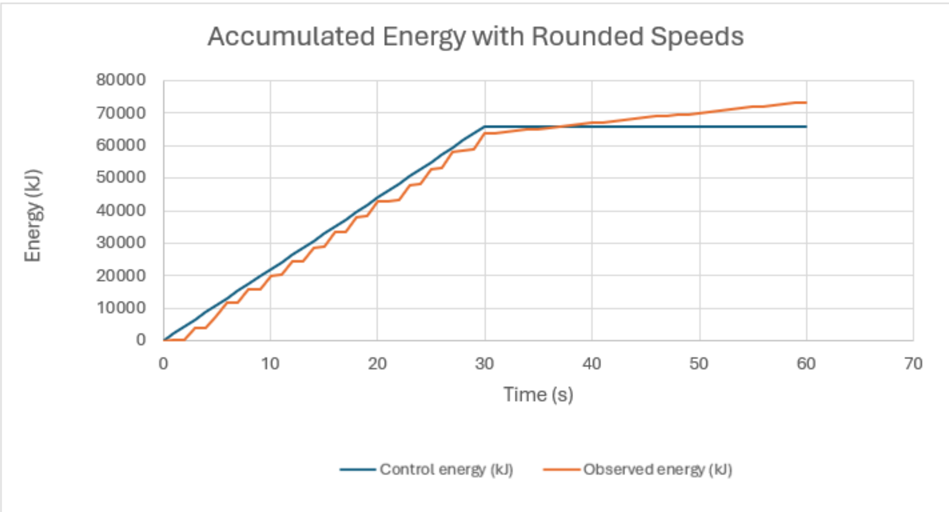


Figure 3.5: Energy profile for rounded speed observations.

For the first 30 seconds, when speed is increasing, we should have a smooth increase in energy, and then no change in energy when the train is coasting, as shown by the blue energy profile. However, the rounded speeds cause steps in the energy estimation during the power phase and consistent overestimation during the coast phase, as shown by the orange curve. During the coast phase, each period where the observed train speed remains constant has a corresponding energy use greater than zero, and each time the speed drops by 1 km/h the energy use is estimated to be zero. For trains without regenerative braking, Energymiser is biased towards overestimating energy because negative power on an interval is ignored.

A key aim of this project was to investigate other methods for estimating energy that may be more reliable.

3.2.3 Energymiser energy estimates during coasting

Energymiser should calculate zero increase in energy use during coasting, but we have shown that inaccuracies in energy estimations can occur due to noisy speed measurements during the coasting phase. To investigate further, we wrote a Python script that calculates the mean power calculated by Energymiser for all time spent in notch 0. The power should be small compared to the maximum power of 2200 kW. Our results, shown in Table 3.3, show that many journeys have significant overestimates of coasting power.

Table 3.3: Mean power for notch 0.

Log	Power (kW)
20230720_001308_212_KR-DL9008B_1	36.35
20230720_001308_212_KR-DL9008B_2	205.45
20230720_004420_620_KR-DF7173_1	168.60
20240130_073151_140_KR-DL9291B_1	274.96
20240130_101838_141_KR-DL9291B_1	31.40
20240130_220634_144_KR-DL9291B_1	31.87
20240201_081944_MP6_KR-DL9291A_1	191.69
20240201_081944_MP6_KR-DL9291A_2	202.23
20240209_114205_MP3_KR-DL9291B_1	468.57
20240209_114205_MP3_KR-DL9291B_2	354.77
20240209_191900_MP10_KR-DL9291A_1	330.61
20240209_191900_MP10_KR-DL9291A_2	475.98
20240210_054351_MP1_KR-DL9291B_1	6.34
20240210_054351_MP1_KR-DL9291B_2	153.99
20240210_161266_MP20_KR-DL9291A_a_1	206.82
20240210_161266_MP20_KR-DL9291A_b_1	289.73
20240213_185702_521_KR-DL9291B_1	530.86
20240213_185702_521_KR-DL9291B_2	227.28
20240214_125404_535_KR-DL9291B_a_1	391.77
20240214_125404_535_KR-DL9291B_a_2	22.41
20240214_125404_535_KR-DL9291B_b_1	174.60
20240219_200352_215_KR-DL9291B_1	407.61
20240221_113610_390_KR-DL9291A_1	42.26

Continued on next page

Table 3.3—*Continued from previous page*

Log	Power (kW)
20240221_182358_391_KR-DL9291B_1	96.80
20240222_020159_328_KR-DL9291A_1	2.23
20240223_014666_566_KR-DL9291A_a_1	176.31
20240223_014666_566_KR-DL9291A_b_1	266.58
20240223_124351_565_KR-DL9291B_1	179.32
20240224_035366_234_KR-DL9291A_a_1	292.72
20240224_035366_234_KR-DL9291A_b_1	383.65
20240224_035366_234_KR-DL9291A_b_2	221.25
20240224_035366_234_KR-DL9291A_c_1	228.75
20240226_015821_MP9_KR-DL9291B_1	229.34
20240226_015821_MP9_KR-DL9291B_2	54.60
20240226_115932_MP8_KR-DL9291A_1	174.65
20240227_003741_183_KR-DL9291B_1	106.77
20240227_081322_321_KR-DL9291B_1	72.56
20240227_115742_MP8_KR-DL9291A_1	0.30
20240228_170466_212_KR-DL9291A_a_1	559.60
20240228_170466_212_KR-DL9291A_b_1	702.76
20240228_170466_212_KR-DL9291A_b_2	380.41
20240313_203920_541_KR-DL9475A_1	287.21
20240314_001159_228_KR-DL9538A_1	1123.28
20240314_001159_228_KR-DL9538A_2	388.40
20240314_001159_228_KR-DL9538A_3	181.77
20240314_001159_228_KR-DL9538A_4	416.53
20240314_014931_391_KR-DL9072B_1	375.57
20240314_014931_391_KR-DL9072B_2	41.55
20240314_014931_391_KR-DL9072B_3	248.02
20240314_014931_391_KR-DL9072B_4	264.76

3.3 Estimating energy from time-in-notch

Instead of using speed measurements to estimate energy use, we can use notch information from the locomotive logs. Energy can be calculated using

$$E_{\text{tin}} = \sum p(n_i)\Delta t_i$$

Figure 3.6: Time-in-notch energy against Energymiser energy.

where n_i is the notch setting at the start of time interval i, $p(i)$ is the corresponding power at the wheels from Table 2.3, and Δt_i is the duration of time interval i. This gives an estimate of the total mechanical energy applied at the wheels for each journey.

At low speeds it is not be possible for a locomotive to apply all of the power available from a notch to the wheels. For example, DL locomotives limit the power at the wheels for speeds less than about 35 km/h. We will ignore this complexity, since most of each journey will be at higher speeds, and lower power notches are usually used when starting.

We can compare the energy calculated from time-in-notch to energy calculated from speed by Energymiser. Figure 3.6 shows the cumulative energy, power, speed, and notch setting for an example journey, with time on the horizontal axis.

The top graph shows the energy calculated from time-in-notch (orange curve) and the energy calculated by Energymiser from speed (blue curve). To the left of the red box, the notch setting is high and the change in energy estimated from the change in speed on each interval (second graph) is noisy, but high. Overestimates of energy are followed by underestimates of energy, and the errors cancel each other out. The cumulative energy estimated from speed is similar to the energy estimated from time-in-notch.

In the red box, the notch settings are lower and we get some noisy energy estimates that are low and negative. The negative values are discarded because the train does not have regenerative braking, so the cumulative energy is biased towards to high energy estimates, which is why Energymiser shows overestimates in periods of coasting.

The time-in-notch method is a more accurate calculation of energy than Energymiser, as it does not include coasting overestimations, inaccuracies caused by GPS inconsistencies, or the rounding of data.

3.4 Smoothing filters

We have shown that errors in speed observations can cause energy use to be overestimated. These errors could be reduced by employing moving average or weighted moving average smoothers to the speed observations before estimating energy.

After a journey, we can consider observations from both sides of each speed observation, resulting in better accuracy compared to solely relying on past points.

A 3-point moving average smoother calculates the smoothed observation at time j as

$$S_j = \frac{Y_{j-1} + Y_j + Y_{j+1}}{3}$$

where Y_j is the raw observation at time j.

Figure 3.7 shows speed observations (grey points) and the smoothed speed profile (blue line) from a 3-point moving average. There is a spike in speed at around 860 seconds into the journey, and the estimated speed still follows the observations.

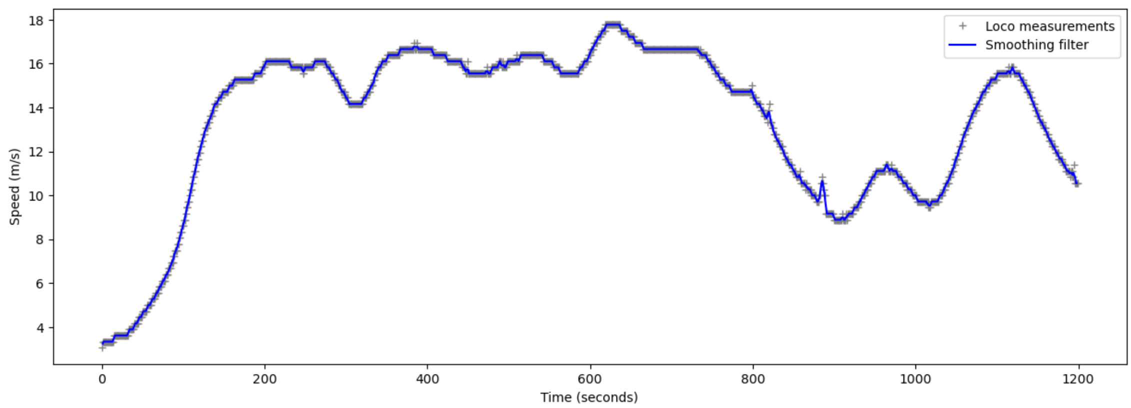


Figure 3.7: Smoothed speeds using a 3 point-moving average.

Figure 3.8 shows detail around this spike. The smoothed speed follows the observation speed closely at the spike. The smoothed profile also has steps due to the rounded speed observations.

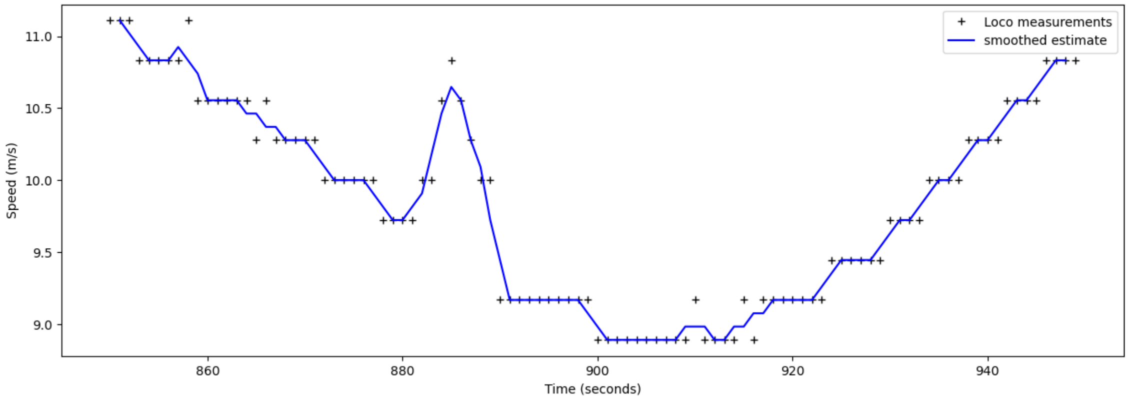


Figure 3.8: Smoothing Filter 3 point-moving average section 850-950 seconds.

Using a 7 point-moving average gives a smoother speed estimate as shown in Figure 3.9. A 7-point moving average smoother uses

$$S_j = \frac{Y_{j-3} + \cdots + Y_{j+3}}{7}.$$

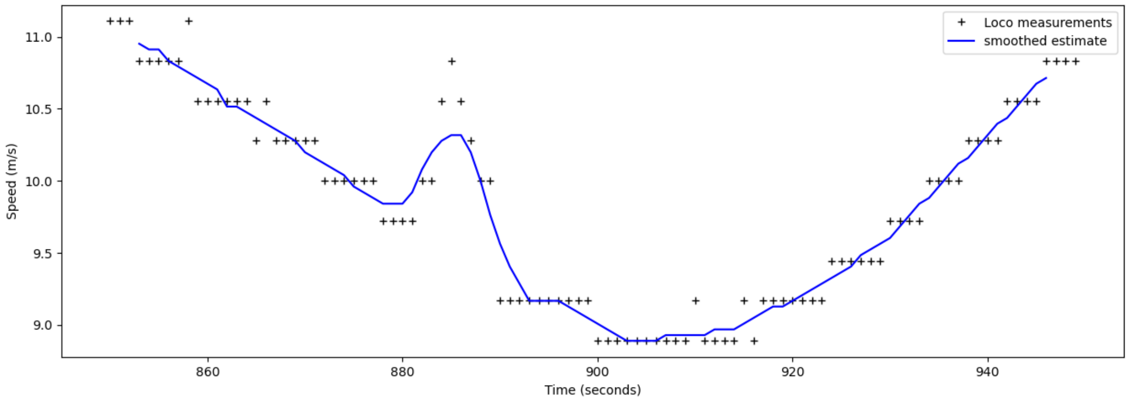


Figure 3.9: Smoothing Filter 7 point-moving average section 850-950 seconds.

Although smoothing filters provide smoother speed estimate, they do not take into account the train dynamics. Section 4 discusses Kalman Filter methods, which take into account system dynamics as well as state observations.

3.5 Estimating energy from sampled speeds

One simple way to reduce the noise in energy estimates is to calculate the change in energy over longer time intervals. To do this we wrote Python code that:

- gets the elevation profile
- gets the log
- samples the log at 10 second intervals
- calculates the gradient force at each location
- calculates the average power on each interval
- accumulates the energy.

We calculated time-in-notch energy profiles for each of the logs, and will use these as a baseline for evaluating other methods of calculating energy.

3.6 Comparing energy estimates

Figure 3.10, which spans many pages at the end of this section, shows three energy profiles for each journey:

- Energymiser calculation (grey)
- energy from sampled speeds (red)
- energy from time-in-notch (green).

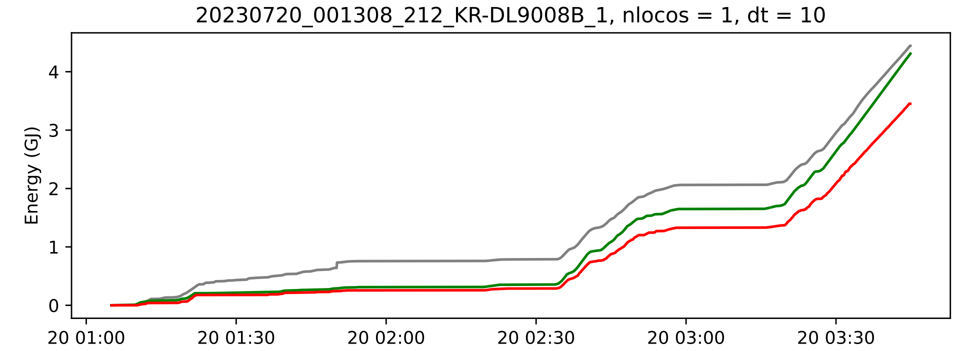
The graphs are in order of timestamp.

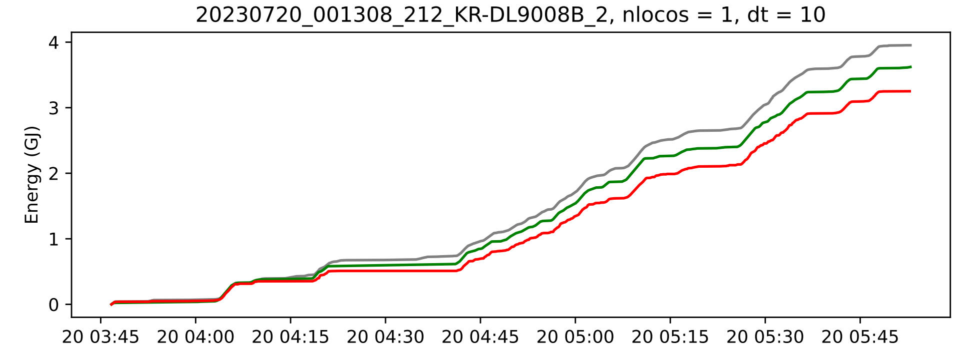
Energy estimated by Energymiser is higher than the other two estimates for all journeys, though for some cases it is not much higher.

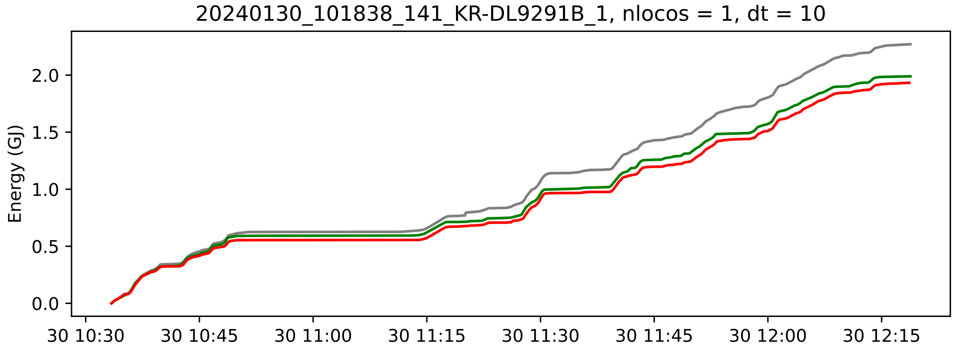
There are some journeys with large differences between the energy profiles. Each of these should be investigated. For some, such as `20240219_200352_215_KR-DL9291B_1`, the notch settings in the locomotive log stopped updating even though the train was still driving.

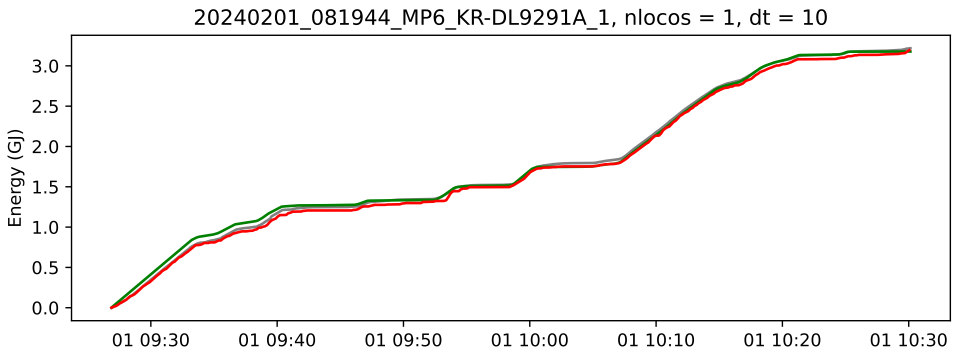
3.7 Conclusion

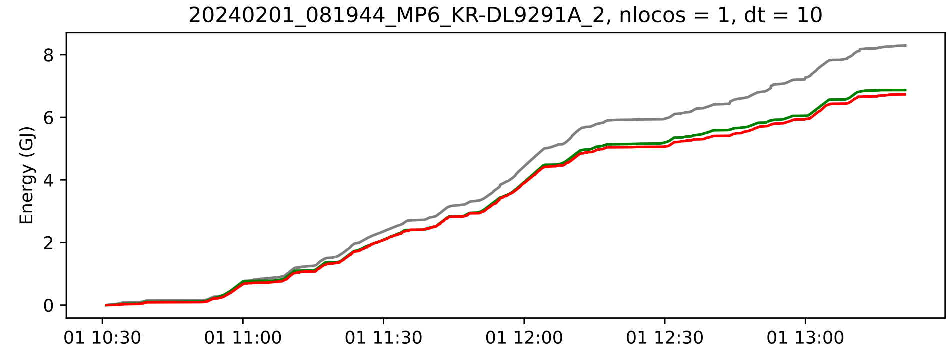
We have shown that even small errors in speed can cause Energymiser to overestimate the energy use, particularly during intervals with low notch settings. If notch data is not available then calculating energy from speeds sampled at 10-second intervals gives better estimates of energy.

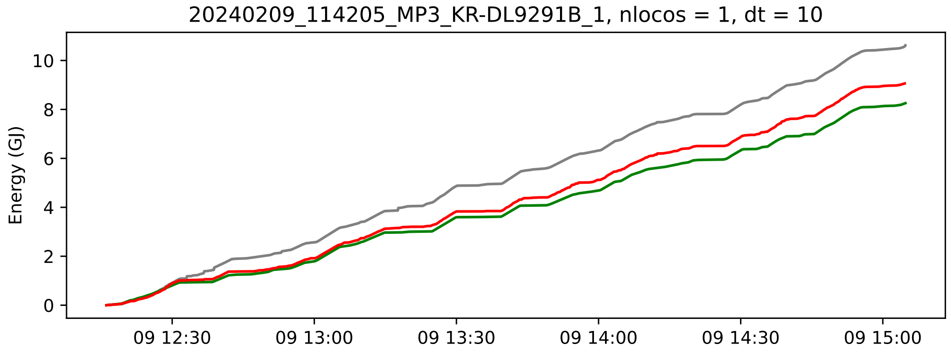

20230720_001308_212_KR-DL9008B_1, nlocos = 1, dt = 10
Energy (GJ)
0
1
2
3
4
20 01:00
20 01:30
20 02:00
20 02:30
20 03:00
20 03:30

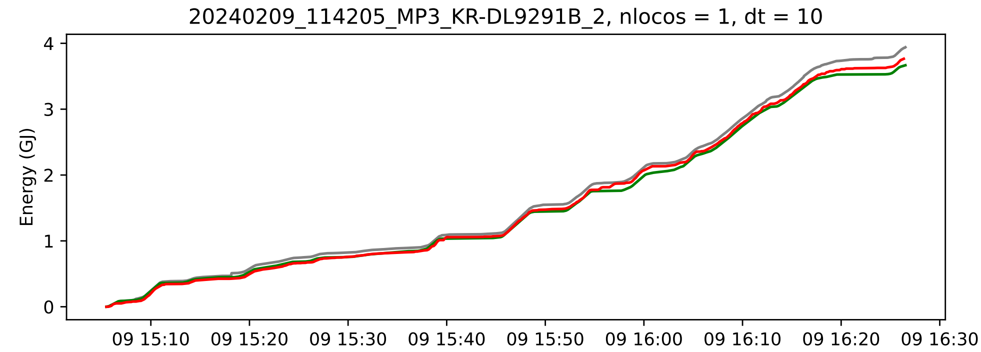

20230720_001308_212_KR-DL9008B_2, nlocos = 1, dt = 10
Energy (GJ)
0
1
2
3
4
20 03:45
20 04:00
20 04:15
20 04:30
20 04:45
20 05:00
20 05:15
20 05:30
20 05:45

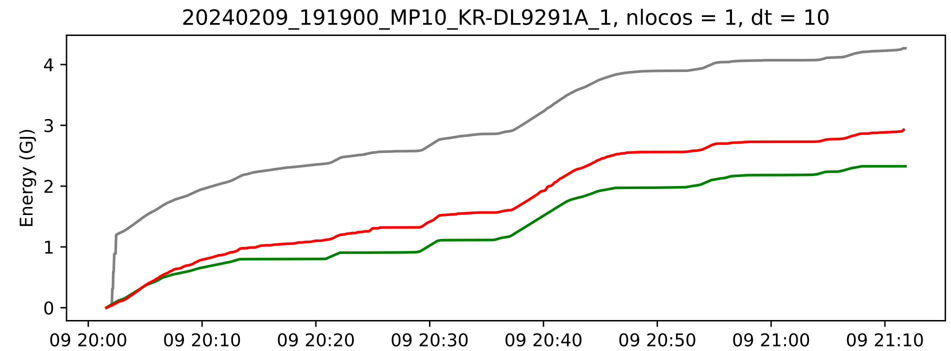

20240130_101838_141_KR-DL9291B_1, nlocos = 1, dt = 10
Energy (GJ)
0.0
0.5
1.0
1.5
2.0
30 10:30
30 10:45
30 11:00
30 11:15
30 11:30
30 11:45
30 12:00
30 12:15

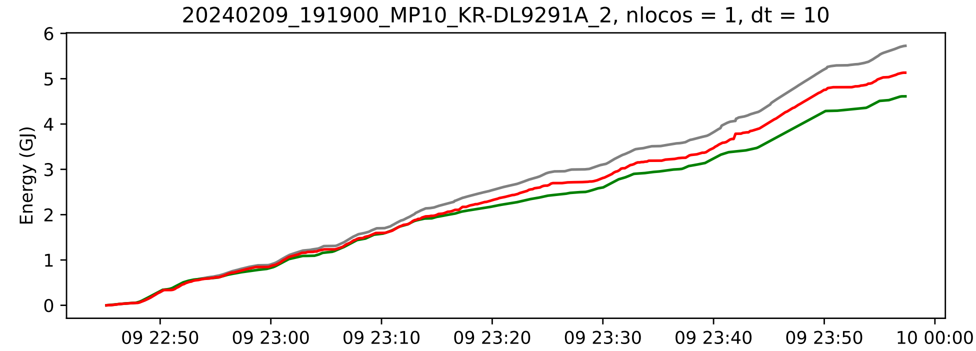

20240201_081944_MP6_KR-DL9291A_1, nlocos = 1, dt = 10
Energy (GJ)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
01 09:30
01 09:40
01 09:50
01 10:00
01 10:10
01 10:20
01 10:30

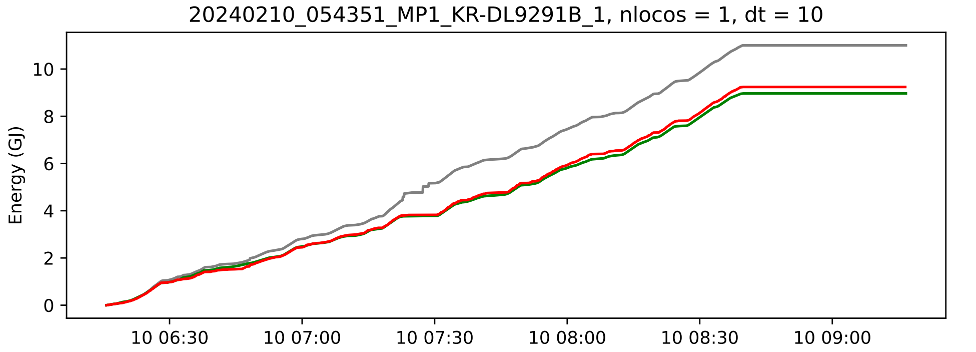

20240201_081944_MP6_KR-DL9291A_2, nlocos = 1, dt = 10
Energy (GJ)
0
2
4
6
8
01 10:30
01 11:00
01 11:30
01 12:00
01 12:30
01 13:00

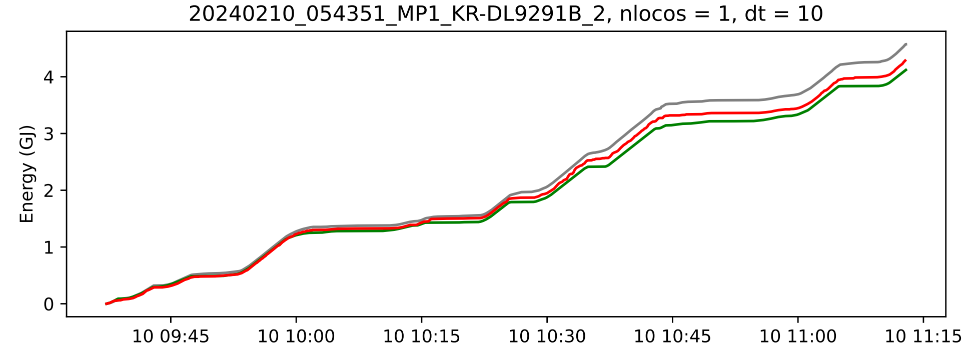

20240209_114205_MP3_KR-DL9291B_1, nlocos = 1, dt = 10
Energy (GJ)
0
2
4
6
8
10
09 12:30
09 13:00
09 13:30
09 14:00
09 14:30
09 15:00

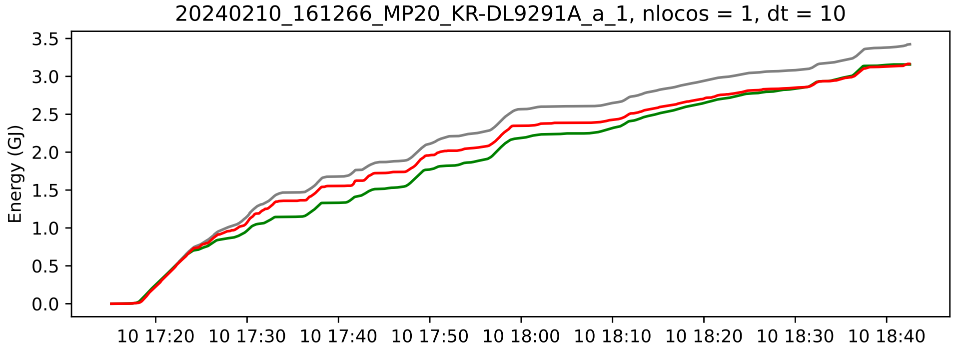

20240209_114205_MP3_KR-DL9291B_2, nlocos = 1, dt = 10
Energy (GJ)
0
1
2
3
4
09 15:10
09 15:20
09 15:30
09 15:40
09 15:50
09 16:00
09 16:10
09 16:20
09 16:30

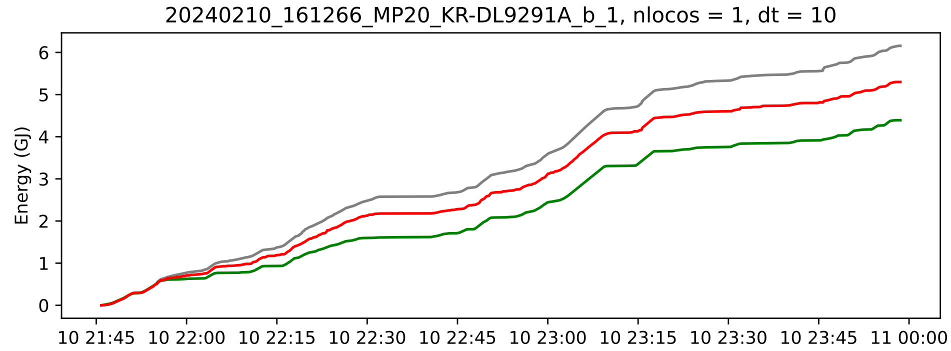

20240209_191900_MP10_KR-DL9291A_1, nlocos = 1, dt = 10
Energy (GJ)
0
1
2
3
4
09 20:00
09 20:10
09 20:20
09 20:30
09 20:40
09 20:50
09 21:00
09 21:10

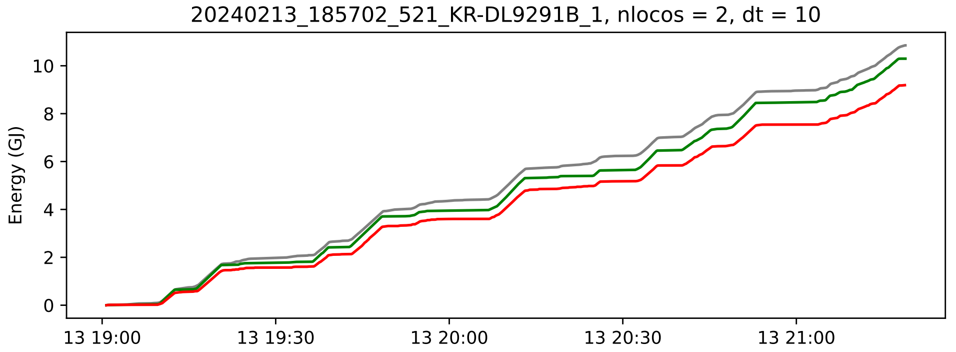

20240209_191900_MP10_KR-DL9291A_2, nlocos = 1, dt = 10
Energy (GJ)
0
1
2
3
4
5
6
09 22:50
09 23:00
09 23:10
09 23:20
09 23:30
09 23:40
09 23:50
10 00:00

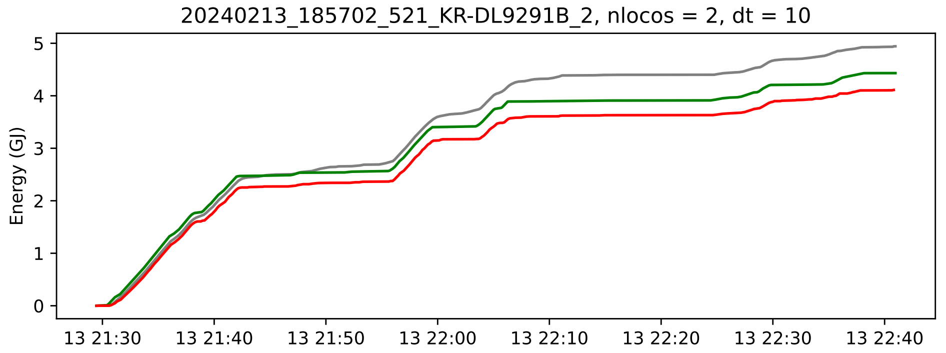

20240210_054351_MP1_KR-DL9291B_1, nlocos = 1, dt = 10
Energy (GJ)
0
2
4
6
8
10
10 06:30
10 07:00
10 07:30
10 08:00
10 08:30
10 09:00

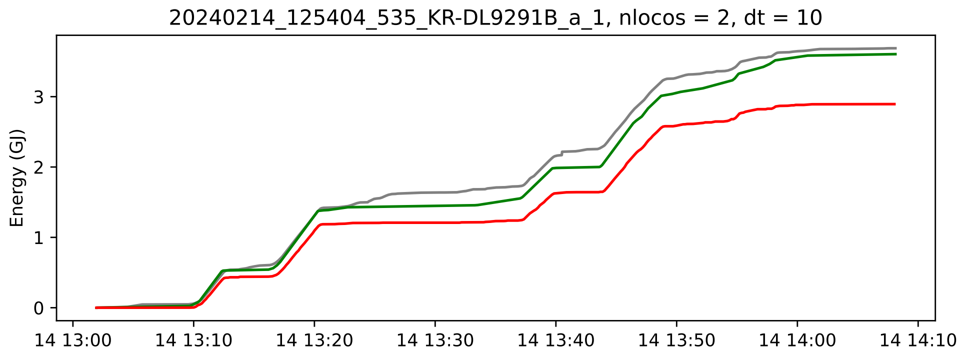

20240210_054351_MP1_KR-DL9291B_2, nlocos = 1, dt = 10
Energy (GJ)
0
1
2
3
4
10 09:45
10 10:00
10 10:15
10 10:30
10 10:45
10 11:00
10 11:15

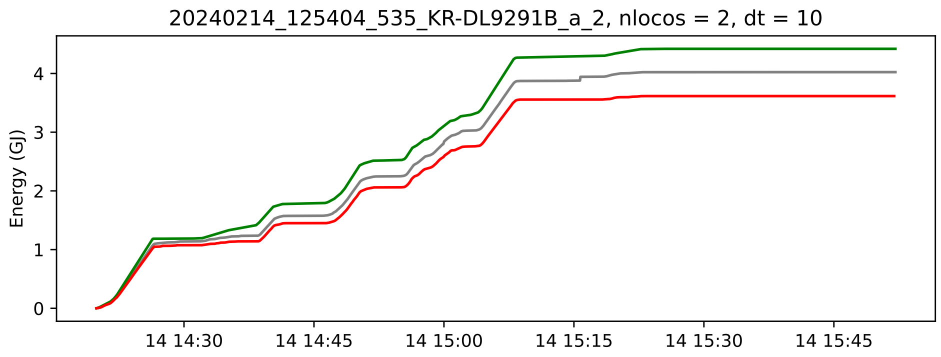

20240210_161266_MP20_KR-DL9291A_a_1, nlocos = 1, dt = 10
Energy (GJ)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
10 17:20
10 17:30
10 17:40
10 17:50
10 18:00
10 18:10
10 18:20
10 18:30
10 18:40

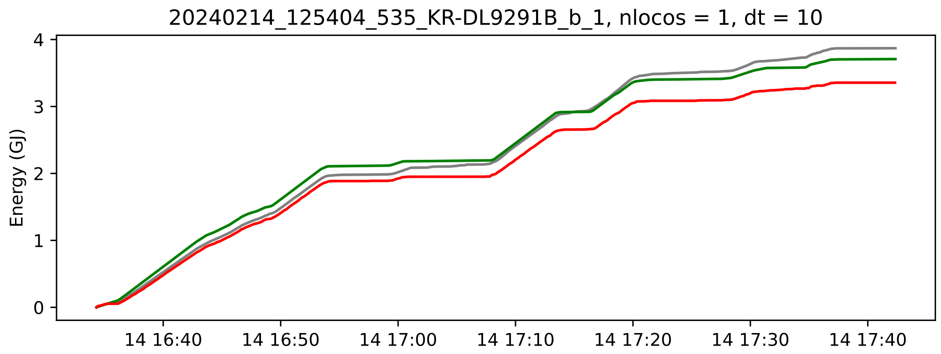

20240210_161266_MP20_KR-DL9291A_b_1, nlocos = 1, dt = 10
Energy (GJ)
10 21:45
10 22:00
10 22:15
10 22:30
10 22:45
10 23:00
10 23:15
10 23:30
10 23:45
11 00:00

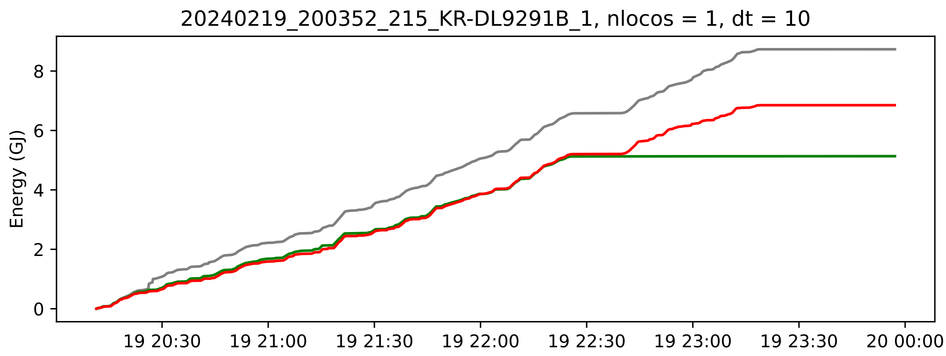

20240213_185702_521_KR-DL9291B_1, nlocos = 2, dt = 10
Energy (GJ)
13 19:00
13 19:30
13 20:00
13 20:30
13 21:00

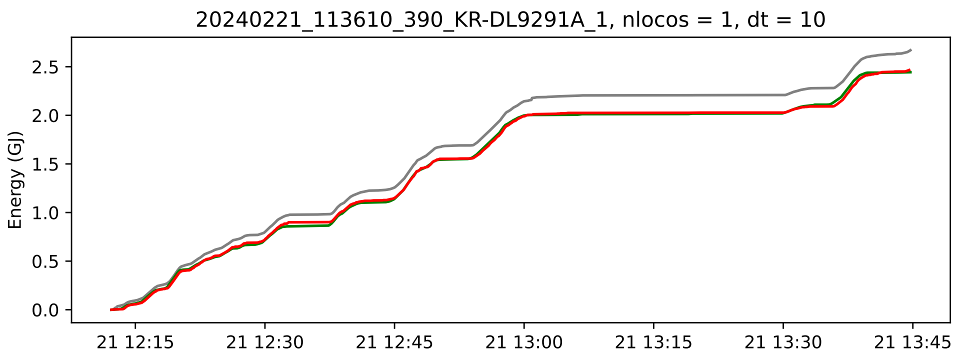

20240213_185702_521_KR-DL9291B_2, nlocos = 2, dt = 10
Energy (GJ)
13 21:30
13 21:40
13 21:50
13 22:00
13 22:10
13 22:20
13 22:30
13 22:40

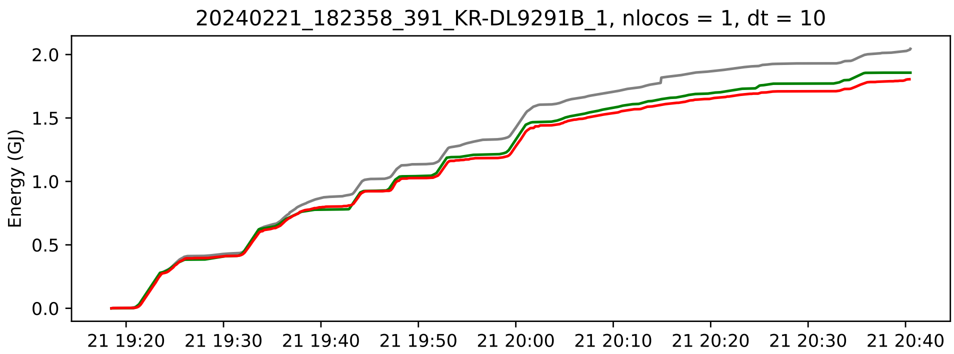

20240214_125404_535_KR-DL9291B_a_1, nlocos = 2, dt = 10
Energy (GJ)
14 13:00
14 13:10
14 13:20
14 13:30
14 13:40
14 13:50
14 14:00
14 14:10

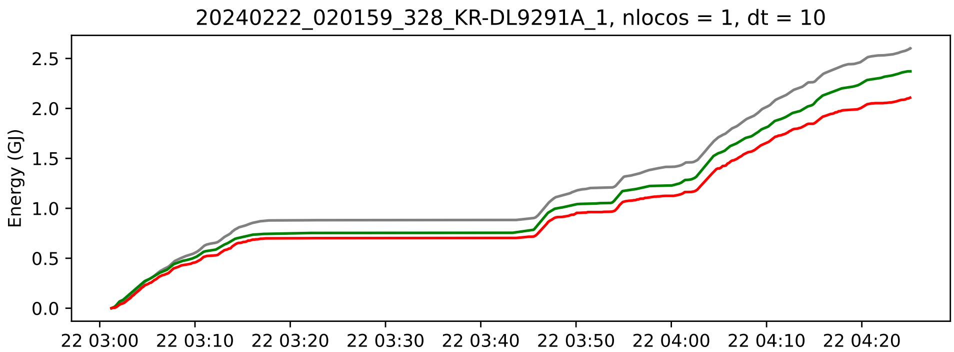

20240214_125404_535_KR-DL9291B_a_2, nlocos = 2, dt = 10
Energy (GJ)
0
1
2
3
4
14 14:30
14 14:45
14 15:00
14 15:15
14 15:30
14 15:45

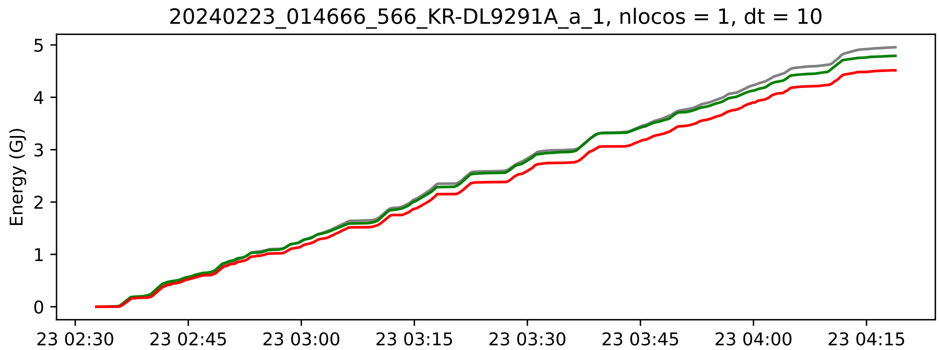

20240214_125404_535_KR-DL9291B_b_1, nlocos = 1, dt = 10
Energy (GJ)
0
1
2
3
4
14 16:40
14 16:50
14 17:00
14 17:10
14 17:20
14 17:30
14 17:40

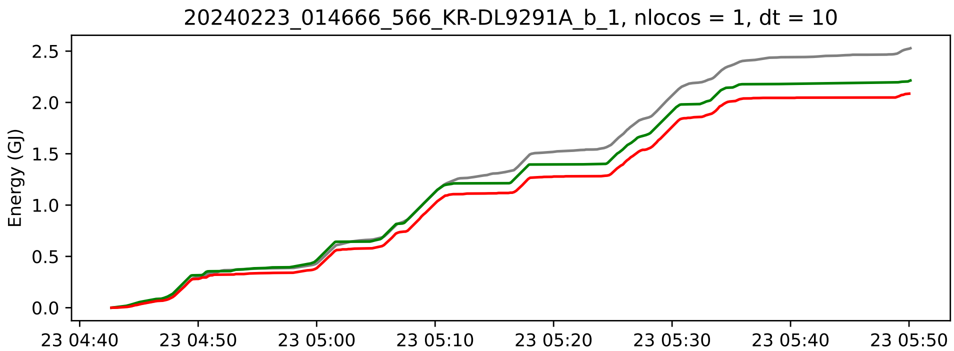

20240219_200352_215_KR-DL9291B_1, nlocos = 1, dt = 10
Energy (GJ)
0
2
4
6
8
19 20:30
19 21:00
19 21:30
19 22:00
19 22:30
19 23:00
19 23:30
20 00:00

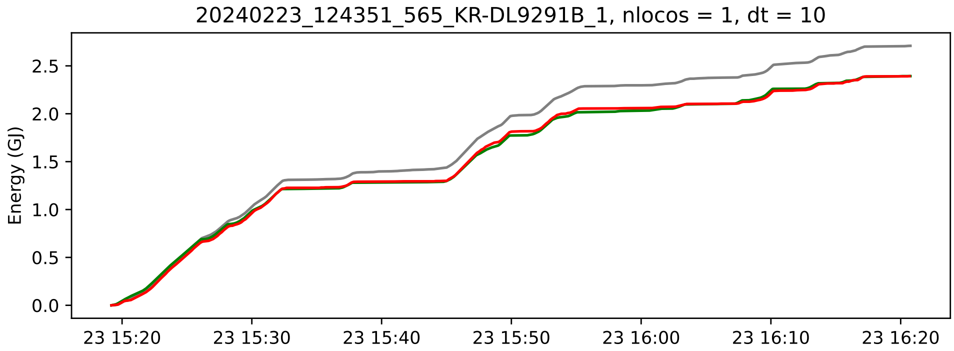

20240221_113610_390_KR-DL9291A_1, nlocos = 1, dt = 10
Energy (GJ)
0.0
0.5
1.0
1.5
2.0
2.5
21 12:15
21 12:30
21 12:45
21 13:00
21 13:15
21 13:30
21 13:45

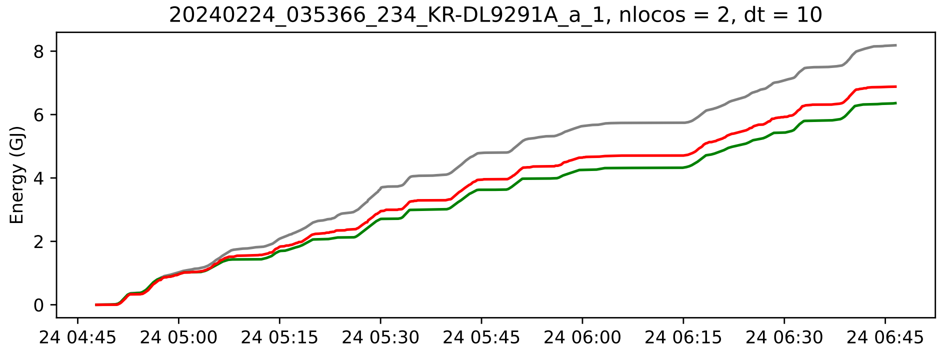

20240221_182358_391_KR-DL9291B_1, nlocos = 1, dt = 10
Energy (GJ)
0.0
0.5
1.0
1.5
2.0
21 19:20
21 19:30
21 19:40
21 19:50
21 20:00
21 20:10
21 20:20
21 20:30
21 20:40

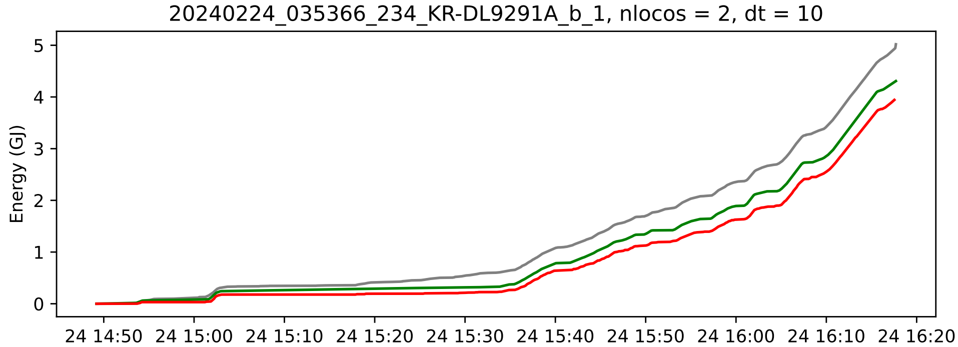

20240222_020159_328_KR-DL9291A_1, nlocos = 1, dt = 10
Energy (GJ)
0.0
0.5
1.0
1.5
2.0
2.5
22 03:00
22 03:10
22 03:20
22 03:30
22 03:40
22 03:50
22 04:00
22 04:10
22 04:20

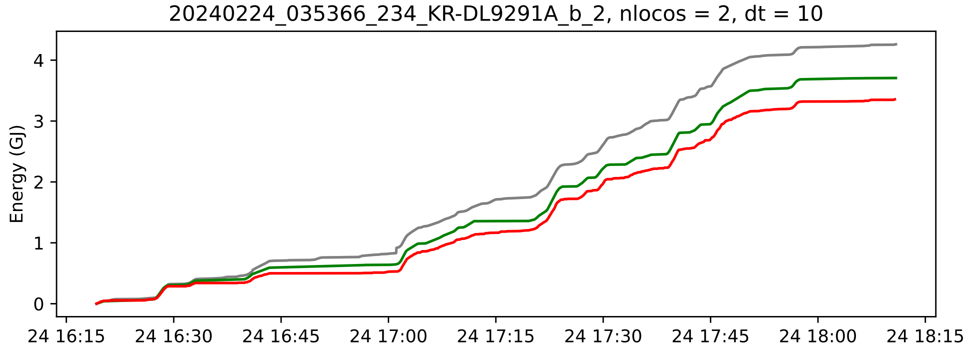

20240223_014666_566_KR-DL9291A_a_1, nlocos = 1, dt = 10
Energy (GJ)
0
1
2
3
4
5
23 02:30
23 02:45
23 03:00
23 03:15
23 03:30
23 03:45
23 04:00
23 04:15


20240223_014666_566_KR-DL9291A_b_1, nlocos = 1, dt = 10
Energy (GJ)
0.0
0.5
1.0
1.5
2.0
2.5
23 04:40
23 04:50
23 05:00
23 05:10
23 05:20
23 05:30
23 05:40
23 05:50

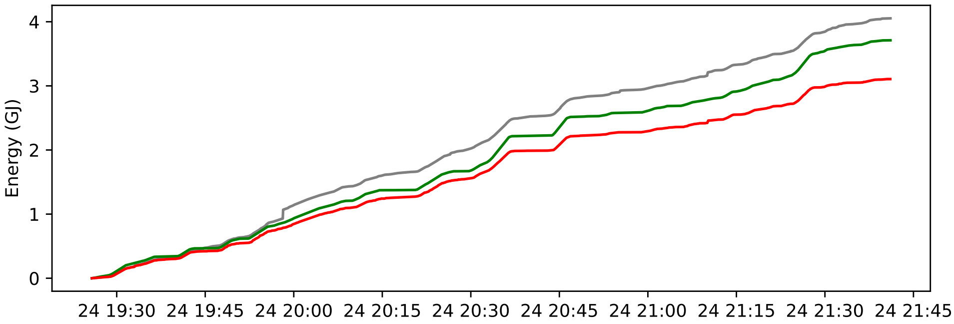

20240223_124351_565_KR-DL9291B_1, nlocos = 1, dt = 10
Energy (GJ)
0.0
0.5
1.0
1.5
2.0
2.5
23 15:20
23 15:30
23 15:40
23 15:50
23 16:00
23 16:10
23 16:20


20240224_035366_234_KR-DL9291A_a_1, nlocos = 2, dt = 10
Energy (GJ)
0
2
4
6
8
24 04:45
24 05:00
24 05:15
24 05:30
24 05:45
24 06:00
24 06:15
24 06:30
24 06:45

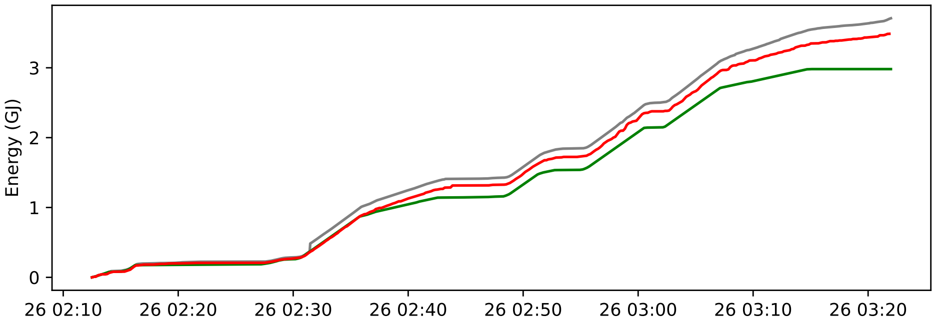

20240224_035366_234_KR-DL9291A_b_1, nlocos = 2, dt = 10
Energy (GJ)
0
1
2
3
4
5
24 14:50
24 15:00
24 15:10
24 15:20
24 15:30
24 15:40
24 15:50
24 16:00
24 16:10
24 16:20


20240224_035366_234_KR-DL9291A_b_2, nlocos = 2, dt = 10
Energy (GJ)
0
1
2
3
4
24 16:15
24 16:30
24 16:45
24 17:00
24 17:15
24 17:30
24 17:45
24 18:00
24 18:15

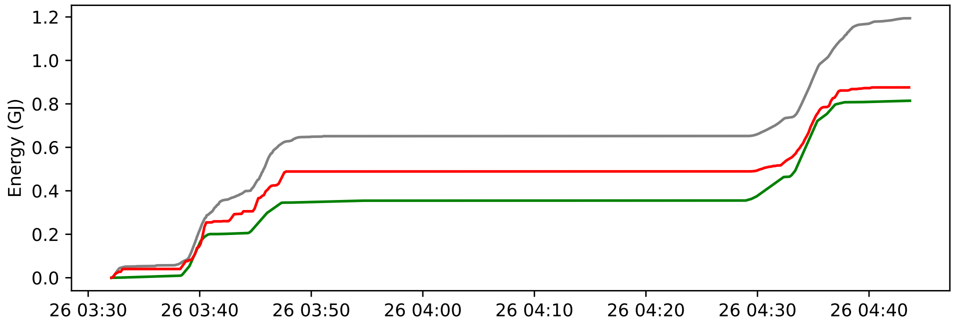
20240224_035366_234_KR-DL9291A_c_1, nlocos = 1, dt = 10



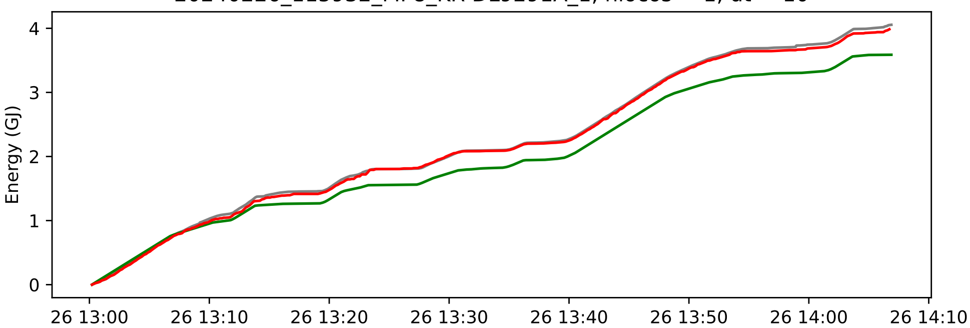
20240226_015821_MP9_KR-DL9291B_1, nlocos = 1, dt = 10



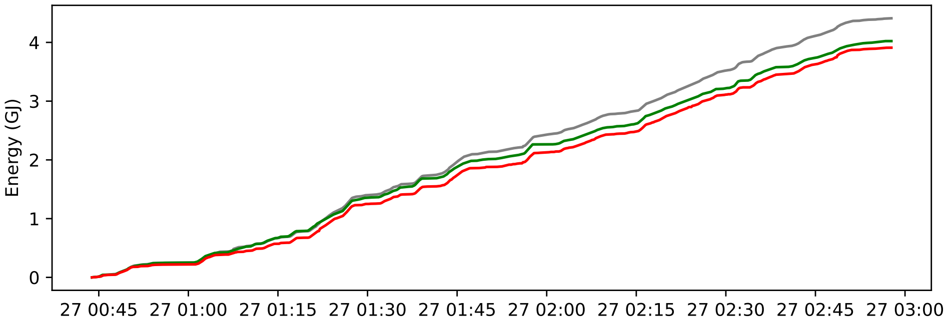
20240226_015821_MP9_KR-DL9291B_2, nlocos = 1, dt = 10



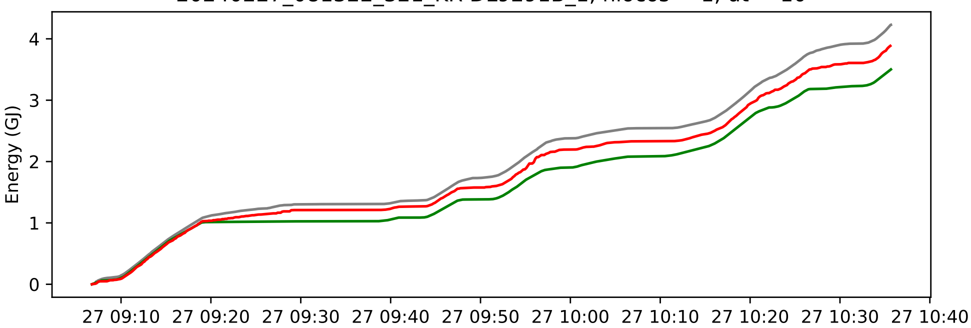
20240226_115932_MP8_KR-DL9291A_1, nlocos = 1, dt = 10



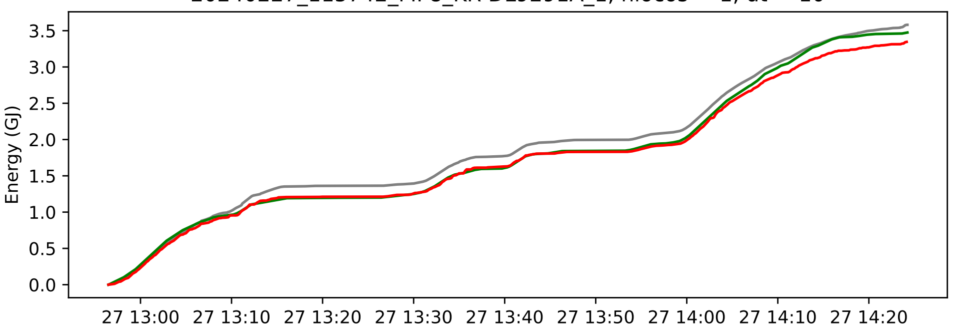
20240227_003741_183_KR-DL9291B_1, nlocos = 1, dt = 10



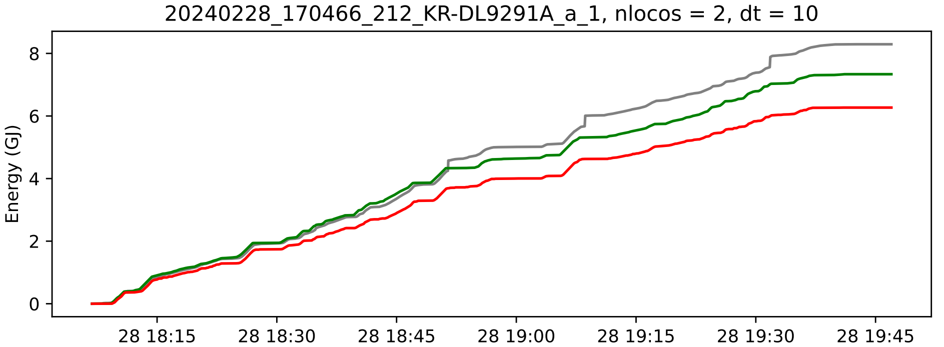
20240227_081322_321_KR-DL9291B_1, nlocos = 1, dt = 10

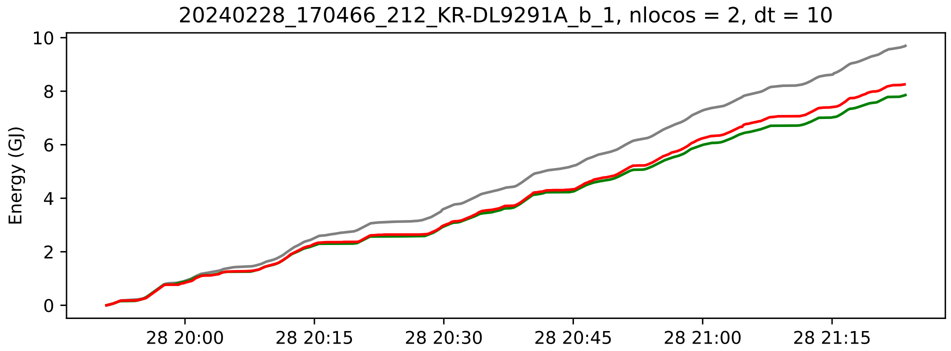


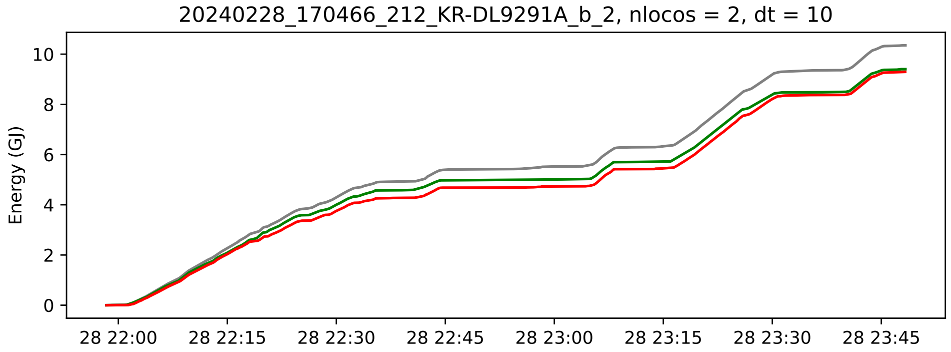
20240227_115742_MP8_KR-DL9291A_1, nlocos = 1, dt = 10

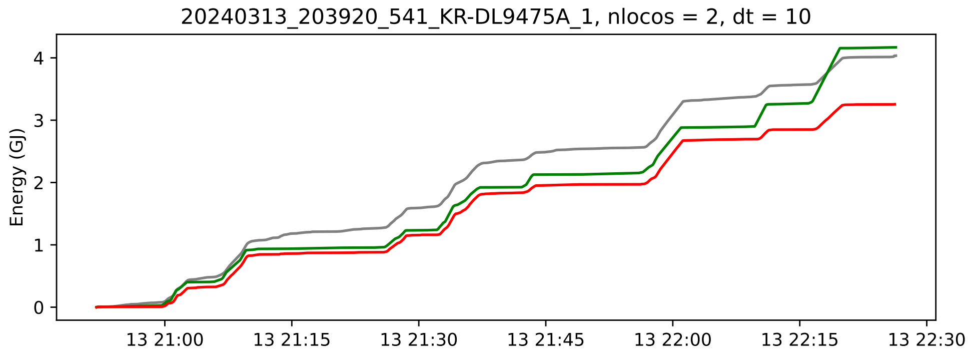


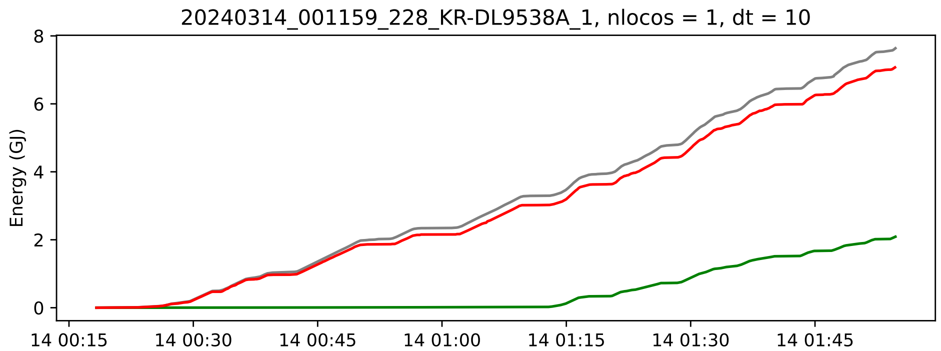
20240228_170466_212_KR-DL9291A_a_1, nlocos = 2, dt = 10

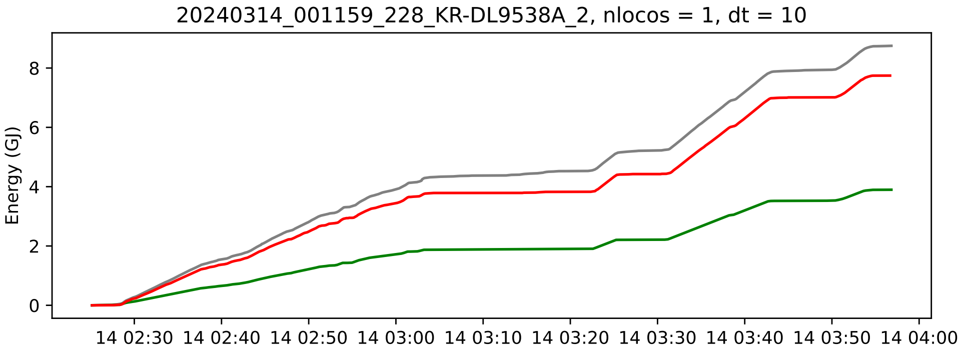


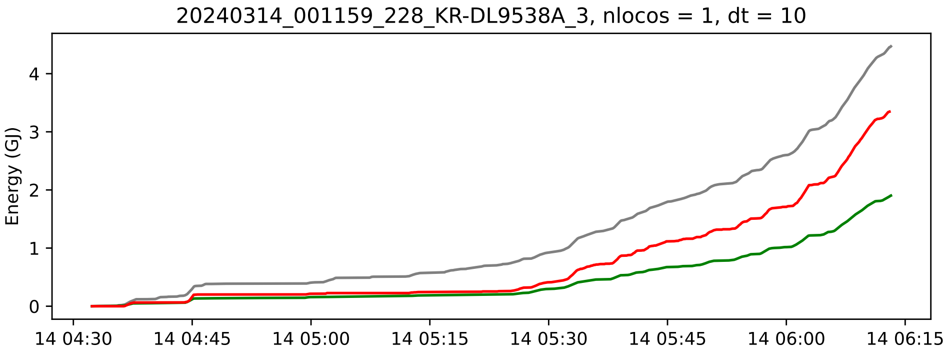

20240228_170466_212_KR-DL9291A_b_1, nlocos = 2, dt = 10
Energy (GJ)
0
2
4
6
8
10
28 20:00
28 20:15
28 20:30
28 20:45
28 21:00
28 21:15

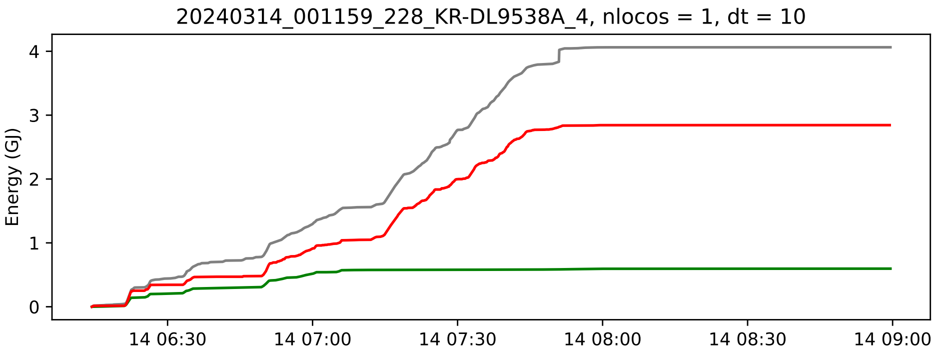

20240228_170466_212_KR-DL9291A_b_2, nlocos = 2, dt = 10
Energy (GJ)
0
2
4
6
8
10
28 22:00
28 22:15
28 22:30
28 22:45
28 23:00
28 23:15
28 23:30
28 23:45

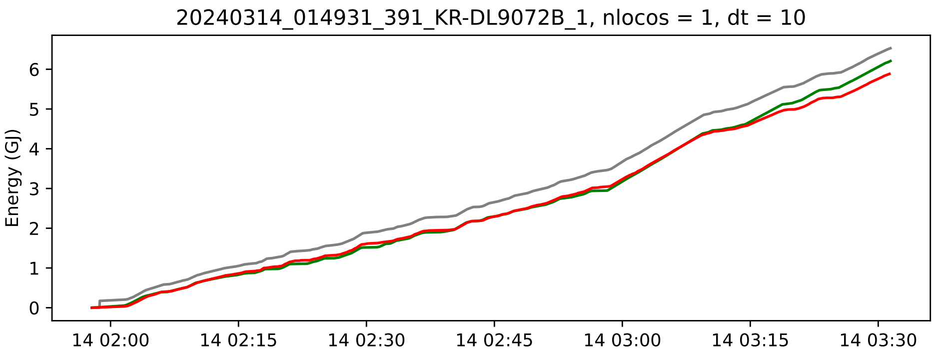

20240313_203920_541_KR-DL9475A_1, nlocos = 2, dt = 10
Energy (GJ)
0
1
2
3
4
13 21:00
13 21:15
13 21:30
13 21:45
13 22:00
13 22:15
13 22:30


20240314_001159_228_KR-DL9538A_1, nlocos = 1, dt = 10
Energy (GJ)
0
2
4
6
8
14 00:15
14 00:30
14 00:45
14 01:00
14 01:15
14 01:30
14 01:45


20240314_001159_228_KR-DL9538A_2, nlocos = 1, dt = 10
Energy (GJ)
0
2
4
6
8
14 02:30
14 02:40
14 02:50
14 03:00
14 03:10
14 03:20
14 03:30
14 03:40
14 03:50
14 04:00


20240314_001159_228_KR-DL9538A_3, nlocos = 1, dt = 10
Energy (GJ)
0
1
2
3
4
14 04:30
14 04:45
14 05:00
14 05:15
14 05:30
14 05:45
14 06:00
14 06:15


20240314_001159_228_KR-DL9538A_4, nlocos = 1, dt = 10
Energy (GJ)
0
1
2
3
4
14 06:30
14 07:00
14 07:30
14 08:00
14 08:30
14 09:00


20240314_014931_391_KR-DL9072B_1, nlocos = 1, dt = 10
Energy (GJ)
0
1
2
3
4
5
6
14 02:00
14 02:15
14 02:30
14 02:45
14 03:00
14 03:15
14 03:30

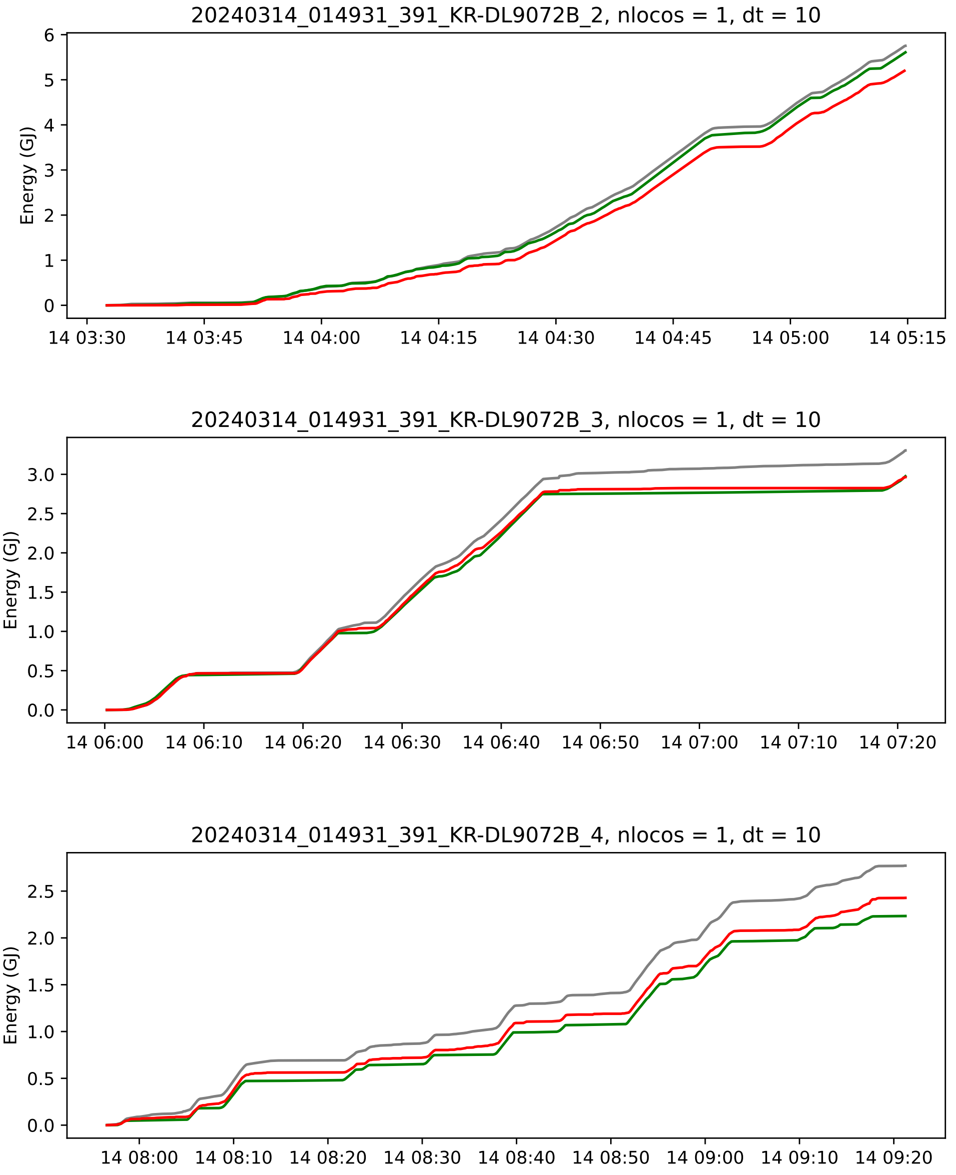


Figure 3.10: Energy calculated by Energymiser (grey), from sampled speeds (green) and from time-in-notch (red).

4 Better energy estimates

Section 3 shows that noisy speed observations can lead to overestimates of energy use. In Section 3.4 we show how smoothing filters can be used to give smoother estimates of speed, which should give better energy estimates. However, simple smoothing filters do not take into account the dynamics of the train.

In this section we develop a mathematical filter that combines locomotive data with Energymiser data to give better estimates of energy use. Once we have got good estimates of energy use using both sets of data, we modify the filter to estimate energy use without requiring locomotive data.

Section 4.1 discusses the Kalman Filter, which combines a mathematical model of the train motion with observations of speed to estimate speed and energy. However, the classic Kalman Filter assumes a linear system model, but our equations of motion are nonlinear. The Unscented Kalman Filter, discussed in Section 4.2, is more appropriate, and easier to implement.

We have implemented our filters using Python code.

4.1 Kalman filters

A Kalman Filter is a method for estimating the state of a dynamic system that, in our case, takes into account the equations of motion for the train. It is a real-time optimal estimator for a linear model with Gaussian noise. It combines predictions from a dynamic model with observed data, adjusting estimates based on measurement noise. Figure 4.1 shows the two steps of the iterative process:

- In the **predict** step the Kalman Filter predicts the state and error covariance ahead for the next time step.
- In the **update** step the filter compares the prediction to the measurements and updates the state estimate and the error covariances using in the prediction steps.

The output from the update step is then looped back into the prediction step, ensuring a continuous refinement of the filter's estimation process.

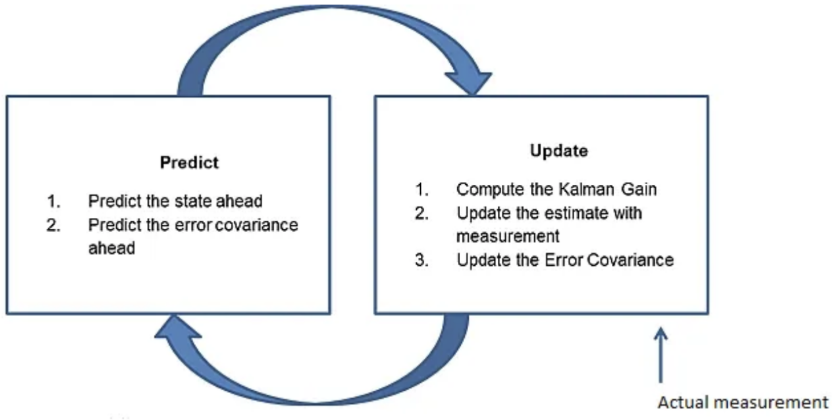


Figure 4.1: Conceptual diagram of a Kalman Filter [7].

The system dynamics of a Kalman Filter are given by an equation with the form

$$x_k = Ax_{k-1} + \omega_{k-1}$$

where x_k is the state of the system at time k, A is a state transition matrix that describes how the state evolves from one time step to the next, and ω_k represents process noise.

A Kalman Filter also incorporates measurements

$$z_k = Hx_k + \nu_k$$

where z_k is the predicted observation at time k, H describes how the observation depends on the state, and ν_k is the observation noise at time k.

However, a Kalman Filter is optimal only for linear systems with Gaussian noise, and our train dynamics are nonlinear. In the next section we discuss the Unscented Kalman Filter, which is designed to overcome these limitations.

4.2 Unscented Kalman Filter

The motion of a train is given by a differential equation

$$m\frac{dv}{dt} = F(u, v) + G(x) - R(v)$$

where m is the mass of the train, $F(u, v)$ is the traction or braking force that depends on control u and speed v, $G(x)$ is the gradient force at location x, and $R(v)$ is the resistance forces acting against the train at speed v [6]. This is a non-linear system because the traction force $F(u, v)$ is non-linear, as we will see in Section 4.3.

An Unscented Kalman Filter (UKF) [3] is an extension of the Kalman Filter designed to deal with non-linear systems through the use of a deterministic sampling approach. In particular, the probability distribution of the state variable is represented by a set of 'sigma' points [4] that are transformed by the nonlinear system equations.

Howlett, Pudney, and Vu [2] describe the use of an Unscented Kalman Filter to estimate coeffiecients of resistance for a train coasting on a flat track.Cunillera et al. [1] use a similar method for estimating resistance coefficients of a train when the tractive and gradient forces are known.

4.3 UKF model with speed, control, gradient, and energy

We implemented an Unscented Kalman Filter using the Python `FilterPy` library [5]. Our filter uses observations of speed, control, and gradient data to estimate energy.

The UKF **state variables** for our system are:

$$x = \begin{pmatrix} v \\ u \\ G \\ E \end{pmatrix} \qquad \begin{array}{l} \text{estimated speed (m/s)} \\ \text{estimated control } [-1, 1] \\ \text{estimated gradient force (kN)} \\ \text{estimated energy (kJ)} \end{array}$$

where v is speed, u is control and G is gradient force and E is energy.

The **state transition** equations are

$$v_k = \begin{cases} v_{k-1} + \frac{1}{m}\left(\frac{P(u_{k-1})}{v_{k-1}} - R(v_{k-1}) + G_{k-1}\right)\Delta t + \omega_v, & \text{if } v_{k-1} > 10 \\ v_{k-1} + \frac{1}{m}\left(\frac{P(u_{k-1})}{10} - R(v_{k-1}) + G_{k-1}\right)\Delta t + \omega_v, & \text{if } v_{k-1} \leq 10 \end{cases}$$

$$u_k = u_{k-1} + \omega_u$$

$$G_k = G_{k-1} + \omega_G$$

$$E_k = \begin{cases} E_{k-1}, & \text{if } u_{k-1} \leq 0.001 \\ E_{k-1} + P(u_{k-1})\Delta t + \omega_E, & \text{if } v_{k-1} > 10 \text{ and } u_{k-1} > 0.001 \\ E_{k-1} + P(u_{k-1})\frac{v_{k-1}}{10}\Delta t + \omega_E, & \text{if } v_{k-1} \leq 10 \text{ and } u_{k-1} > 0.001 \end{cases}$$

where m is the known mass of the train, $P(u)$ is the controlled tractive power, $R(v) = r_0 + r_1 v + r_2 v^2$ is the resistance force with known coefficients r_0, r_1, r_2, G is an known gradient force.

The first equation predicts the next speed based on the current speed, control and gradient. The controlled tractive power is estimated by

$$P(u) = \begin{cases} 1000n(1971u^2 + 341.3u), & \text{if } u \geq 0 \\ 2000nu, & \text{if } u < 0 \end{cases}$$

where n is the number of locomotives. When the control $u \geq 0$ the power is a quadratic function of the control. When $u < 0$ we assume that the braking power is proportional to u.

The second equation predicts that the control will not change. However, the noise term ω_u allows the control to change to follow the observed control.

The third equation predicts that gradient force will not change, but allows the gradient control to change to follow the observed gradient force.

In the energy transition equation we accumulate energy only if $u > 0.001$. We use this small value rather than zero so that sigma points with small u during coast phases do not contribute to energy accumulation—energy is accumulated only if the control u is clearly above the coasting range.

The **observations** are

$$\begin{aligned} y_k &= v_k + \nu_v && \text{speed observation} \\ z_k &= u_k + \nu_u && \text{control observation} \\ w_k &= G_k + \nu_G && \text{gradient force observation.} \end{aligned}$$

Speed and control are observed from the log. Gradient forces are determined from the route data for each journey.

To propagate uncertainty and capture system nonlinearity, the UKF generates **sigma points** around the current state. We used the `MerweScaledSigmaPoints` method for generating and scaling these points, with three parameters:

- α controls the spread of sigma points around the mean
- β controls the shape of the distribution of sigma points
- κ helps fine-tune the sigma point spread about the mean.

4.4 Results

In this section we show UKF results for two example journeys. For each example, we run the UKF in two different modes:

- the control observation noise parameter ν_u is set to a small value so that the filter 'trusts' the observed control
- the control observation noise parameter ν_u is set to a large value so that the control observations are ignored.

The UKF performs better when it has control observations, but these are not always available.

Example 1: UKF trusts the control observations

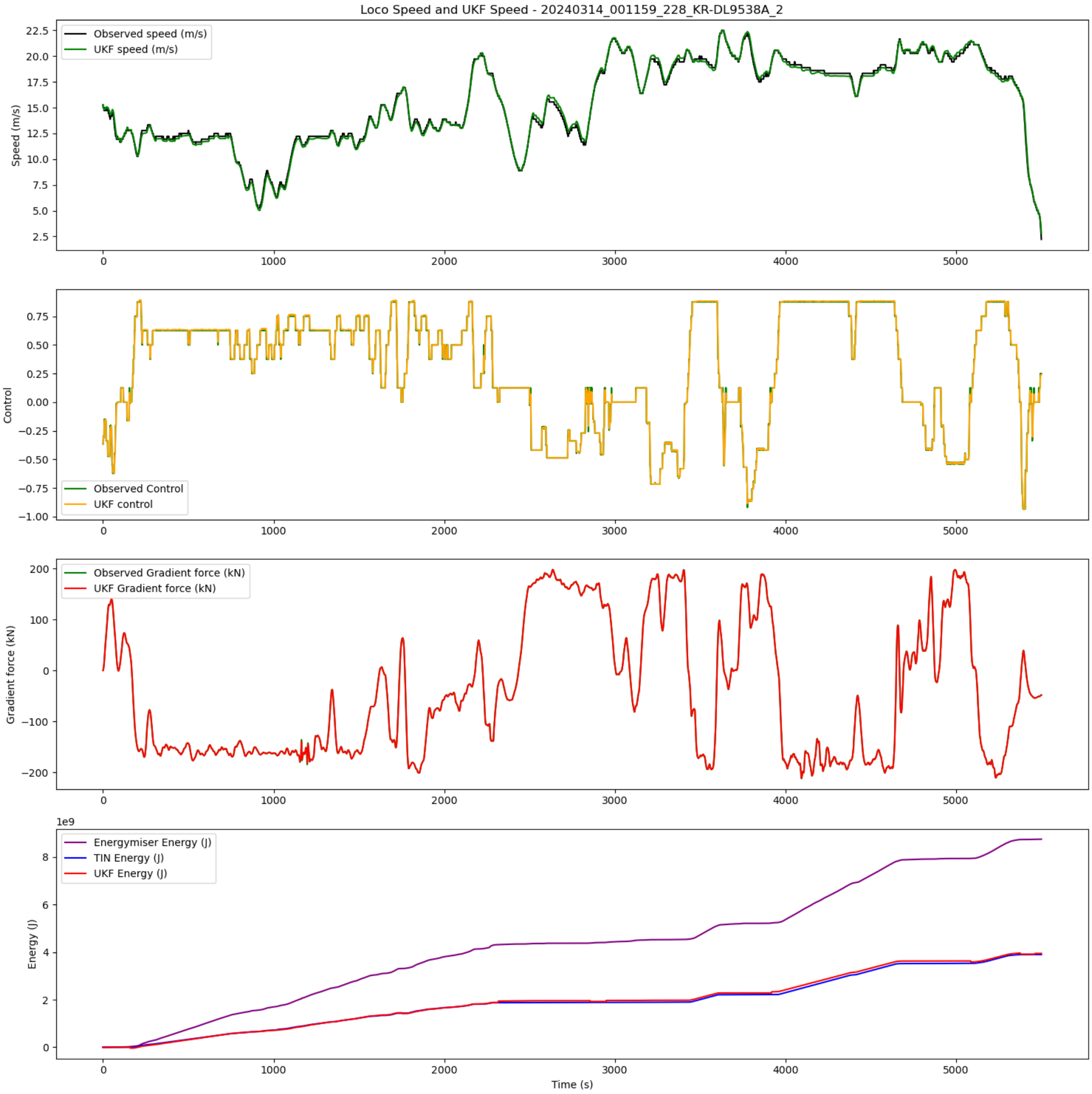


Figure 4.2: Example 1 UKF results using control observations.

The top three graphs show that the UKF speed profile, control and track gradient closely follow the observations.

The bottom graph shows that the UKF energy in red closely aligns with the time-in-notch energy, which is in blue.

Example 1: UKF ignores the control observations

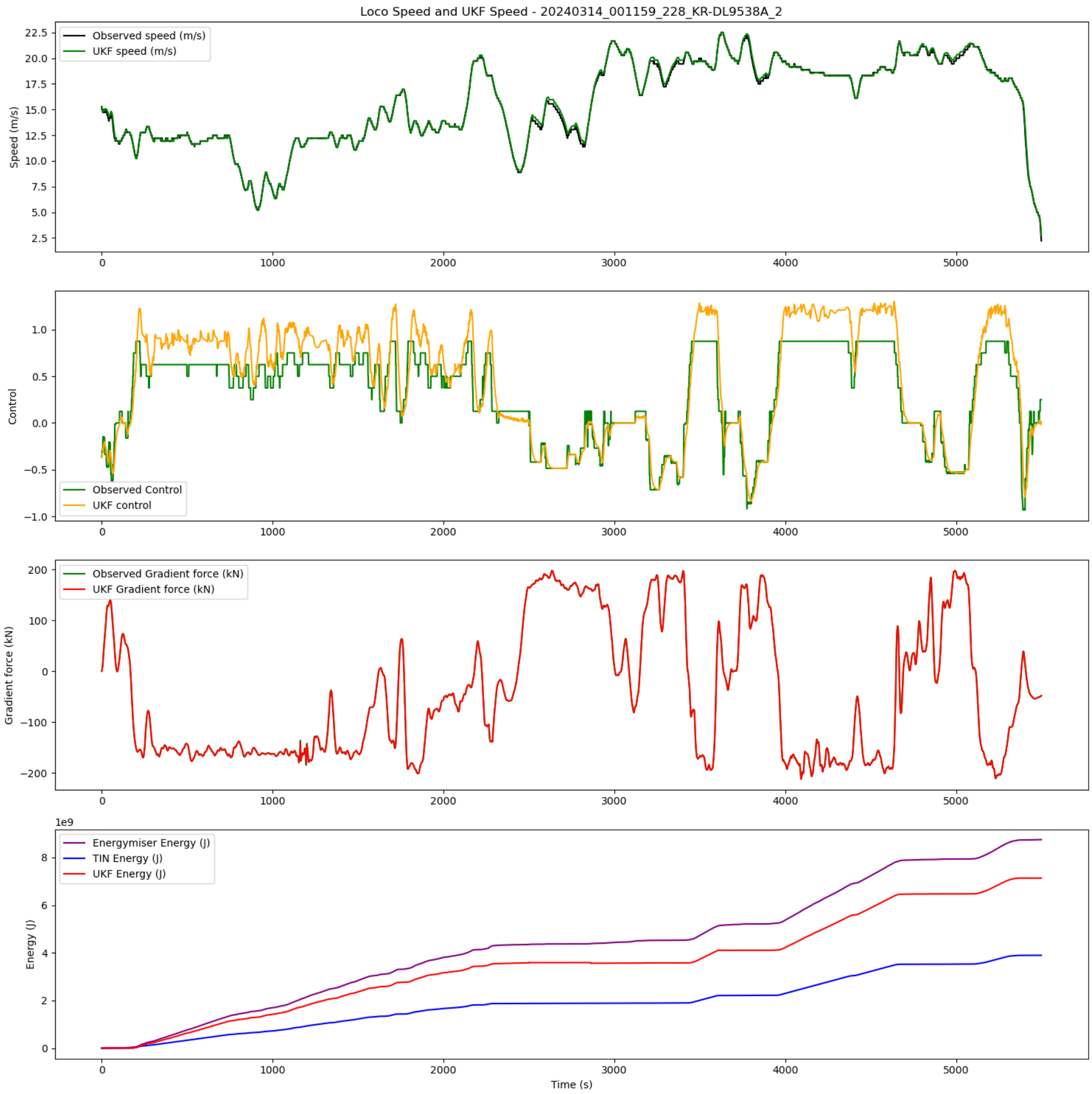


Figure 4.3: Example 1 UKF results ignoring control observations.

This graph displays the results for the same journey when the UKF is set to ignore the control observations.

In this case, the UKF control (in orange) follows the model but fluctuates compared to the observed control (in green).

The bottom graph shows the UKF energy (in red) higher than the time-in-notch Energy (in blue), but lower than the Energymiser Energy.

Example 2: UKF trusts the control observations

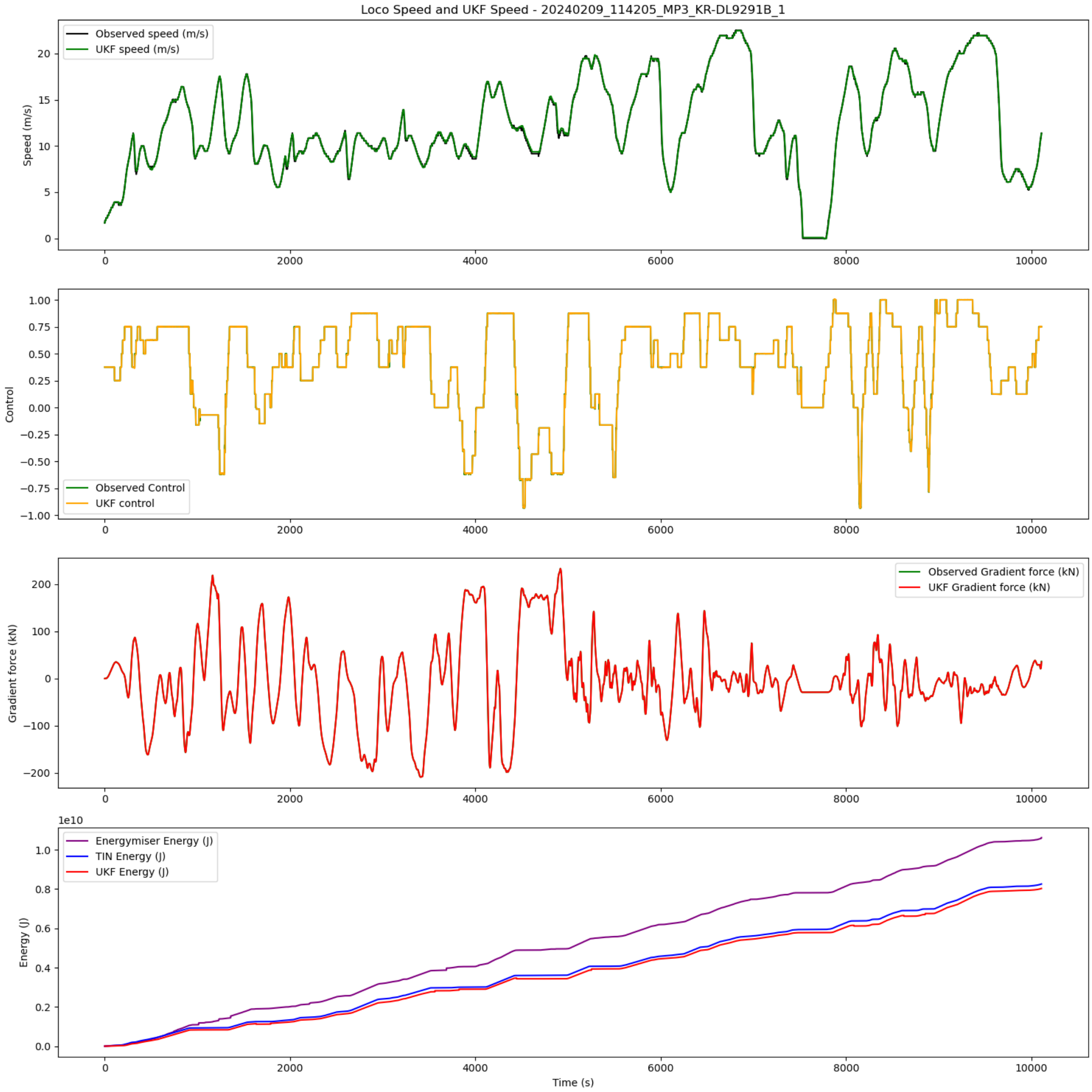


Figure 4.4: Example 2 UKF results using ccontrol observations.

Here are the results from another journey. When using the control observations, the UKF energy estimate closely matches the time-in-notch energy.

Example 2: UKF ignores the control observations

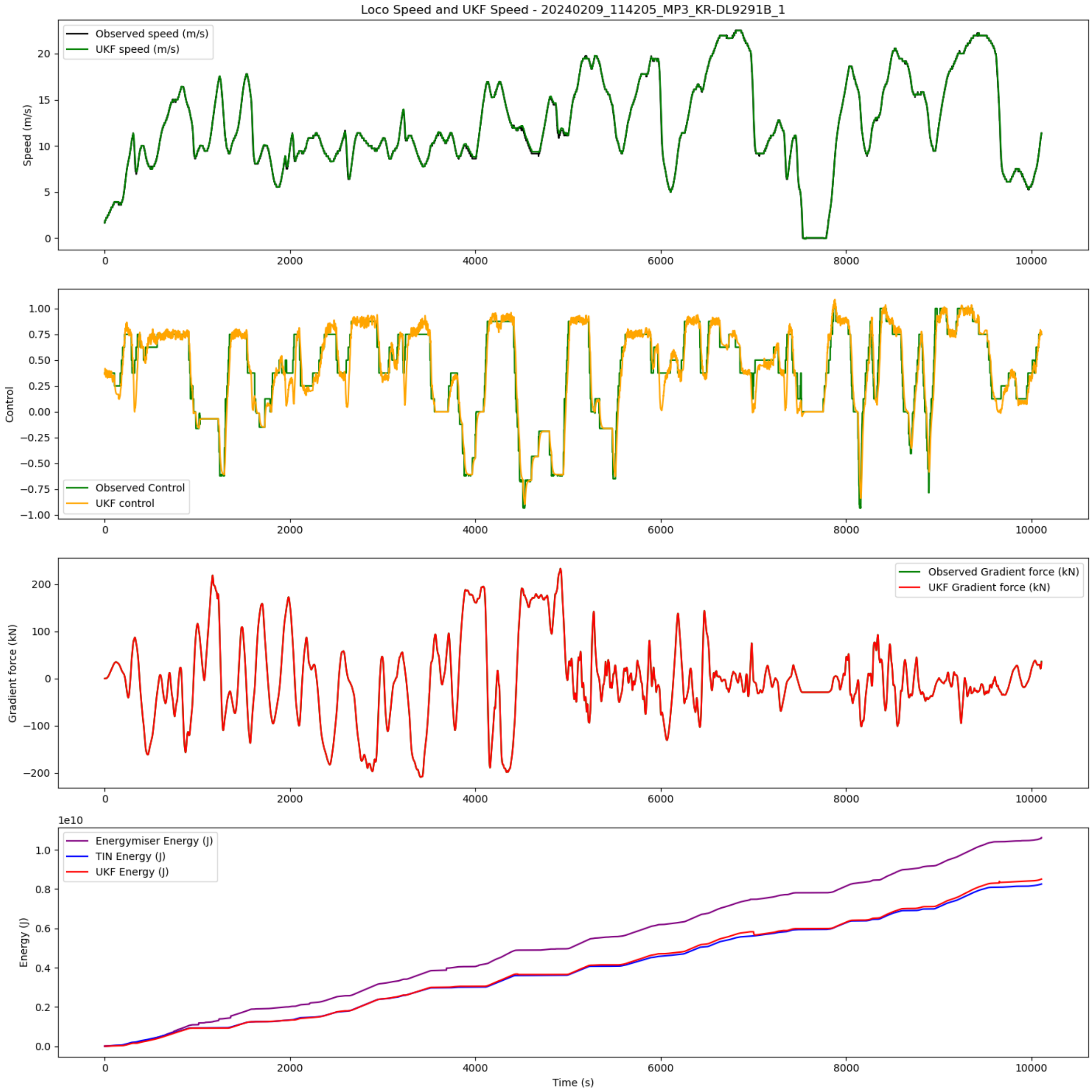


Figure 4.5: Example 2 UKF results ignoring control observations.

Even when ignoring the control observations, the UKF still estimates energy near the time-in-notch energy.

4.5 Comparing energy estimates

Table 4.1 shows energy estimates for each of the journeys using five different methods:

- **E**: Energymiser estimate of energy
- **SS**: energy from sampled speeds
- **TIN**: time-in-notch energy
- **UKF2**: Unscented Kalman Filter without control observations
- **UKF1**: Unscented Kalman Filter with control observations

Table 4.1: Energy use (GJ) for all train journeys using five calculation methods.

Journey	E	SS	TIN	UKF2	UKF1
20230720_001308_212_KR-DL9008B_1	4.44	3.45	4.31	3.71	3.90
20230720_001308_212_KR-DL9008B_2	3.95	3.25	3.62	3.43	3.49
20240130_101838_141_KR-DL9291B_1	2.27	1.93	1.99	2.05	2.13
20240201_081944_MP6_KR-DL9291A_1	3.22	3.19	3.18	3.34	3.33
20240201_081944_MP6_KR-DL9291A_2	8.29	6.73	6.87	6.88	6.90
20240209_114205_MP3_KR-DL9291B_1	10.62	9.06	8.26	8.60	8.24
20240209_114205_MP3_KR-DL9291B_2	3.94	3.77	3.67	3.93	3.87
20240209_191900_MP10_KR-DL9291A_1	4.27	2.93	2.33	2.60	2.36
20240209_191900_MP10_KR-DL9291A_2	5.73	5.13	4.61	5.11	4.91
20240210_054351_MP1_KR-DL9291B_1	11.00	9.24	8.97	9.07	9.17
20240210_054351_MP1_KR-DL9291B_2	4.57	4.28	4.12	4.12	4.02
20240210_161266_MP20_KR-DL9291A_a_1	3.42	3.17	3.16	3.25	3.16
20240210_161266_MP20_KR-DL9291A_b_1	6.16	5.30	4.39	4.99	4.59
20240213_185702_521_KR-DL9291B_1	10.85	9.19	10.30	9.31	9.46
20240213_185702_521_KR-DL9291B_2	4.94	4.11	4.43	4.14	4.31
20240214_125404_535_KR-DL9291B_a_1	3.69	2.89	3.60	3.07	3.35
20240214_125404_535_KR-DL9291B_a_2	4.02	3.62	4.42	3.72	3.89
20240214_125404_535_KR-DL9291B_b_1	3.87	3.36	3.71	3.45	3.74
20240219_200352_215_KR-DL9291B_1	8.73	6.85	5.13	5.98	5.61
20240221_113610_390_KR-DL9291A_1	2.67	2.47	2.44	2.50	2.45
20240221_182358_391_KR-DL9291B_1	2.05	1.81	1.86	1.90	1.86
20240222_020159_328_KR-DL9291A_1	2.60	2.11	2.37	2.25	2.42
20240223_014666_566_KR-DL9291A_a_1	4.96	4.52	4.79	4.70	4.87
20240223_014666_566_KR-DL9291A_b_1	2.53	2.09	2.21	2.12	2.07

Continued on next page

Table 4.1—*Continued from previous page*

Journey	E	SS	TIN	UKF2	UKF1
20240223_124351_565_KR-DL9291B_1	2.71	2.39	2.39	2.46	2.41
20240224_035366_234_KR-DL9291A_a_1	8.19	6.88	6.36	7.09	6.71
20240224_035366_234_KR-DL9291A_b_1	5.02	3.94	4.31	4.34	4.40
20240224_035366_234_KR-DL9291A_b_2	4.26	3.36	3.71	3.87	3.93
20240224_035366_234_KR-DL9291A_c_1	4.05	3.11	3.71	3.45	3.77
20240226_015821_MP9_KR-DL9291B_1	3.71	3.49	2.98	3.33	3.11
20240226_015821_MP9_KR-DL9291B_2	1.19	0.88	0.81	0.83	0.77
20240226_115932_MP8_KR-DL9291A_1	4.06	3.99	3.59	3.79	3.56
20240227_003741_183_KR-DL9291B_1	4.41	3.91	4.02	4.04	4.15
20240227_081322_321_KR-DL9291B_1	4.23	3.88	3.50	3.66	3.40
20240227_115742_MP8_KR-DL9291A_1	3.58	3.35	3.47	3.49	3.55
20240228_170466_212_KR-DL9291A_a_1	8.29	6.27	7.34	6.67	7.11
20240228_170466_212_KR-DL9291A_b_1	9.70	8.25	7.86	8.43	8.15
20240228_170466_212_KR-DL9291A_b_2	10.35	9.29	9.40	9.45	9.33
20240313_203920_541_KR-DL9475A_1	4.03	3.26	4.17	3.10	3.24
20240314_001159_228_KR-DL9538A_1	7.64	7.07	2.09	4.69	2.92
20240314_001159_228_KR-DL9538A_2	8.75	7.75	3.90	7.16	5.06
20240314_001159_228_KR-DL9538A_3	4.47	3.35	1.90	3.19	2.38
20240314_001159_228_KR-DL9538A_4	4.06	2.84	0.60	1.53	1.52
20240314_014931_391_KR-DL9072B_1	6.53	5.88	6.21	5.80	5.98
20240314_014931_391_KR-DL9072B_2	5.75	5.20	5.61	5.21	5.39
20240314_014931_391_KR-DL9072B_3	3.30	2.96	2.98	2.83	2.84
20240314_014931_391_KR-DL9072B_4	2.77	2.43	2.23	2.26	2.20
Mean	**5.19**	**4.43**	**4.21**	**4.36**	**4.26**

The last row summarises the results:

- Energymiser gives the highest average (5.19 GJ). For most journeys, Energymiser produces the highest energy estimates across all methods, suggesting it overestimate energy consumption compared to other techniques.
- Sampled Speed has an average of 4.43 GJ. This method produces lower energy estimates than Energymiser and, on average, slightly higher estimates than the time-in-notch method, indicating that it is a better estimate of energy usage that the current Energymiser method.
- Time-in-notch energy serves as the baseline, with an average of 4.21 GJ.

- UKF2, without control observations has a 4% higher average energy use than the time-in-notch method, at 4.36 GJ. It aligns reasonably well with TIN calculations, often producing similar energy estimates, making it a good option when notch data is unavailable.
- UKF1, with control observations, has an average of 4.26 GJ, which is essentially the same as the time-in-notch estimates.

Figure 4.6 shows how energy estimates from each method (horizontal axis) compare to the time-in-notch energy benchmark. Each point indicates the ratio of the energy estimate for the given method divided by the time-in-notch energy.

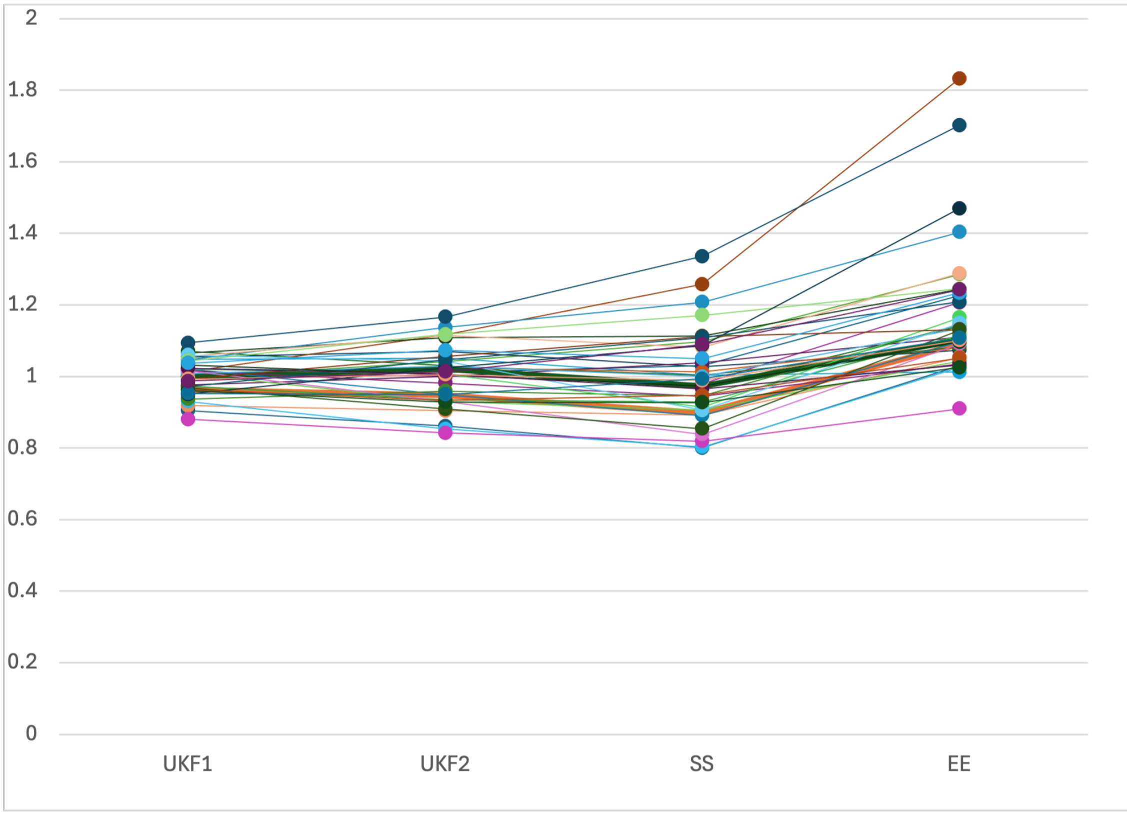


Figure 4.6: Compare Energy Estimates from 4 models to time-in-notch energy.

Each line in the plot connects energy estimates from the same journey. The upward trend across most journeys suggests that:

- UKF with control observations provides the most accurate and consistent estimates relative to the time-in-notch energy benchmark.
- UKF without control observations is the next best performer, aligning reasonably well with TIN energy, though it occasionally provides slightly higher estimates.

- The sampled speed method is somewhat accurate but generally has larger range of energy estimates than the UKF methods.
- Energymiser consistently overestimates energy compared to time-in-notch energy, suggesting a systematic tendency toward higher energy consumption estimates.

To gain deeper insight into the differences between the two UKF models, we compared their results across 48 journey logs, shown in the scatter plot in Figure 4.7 shown.

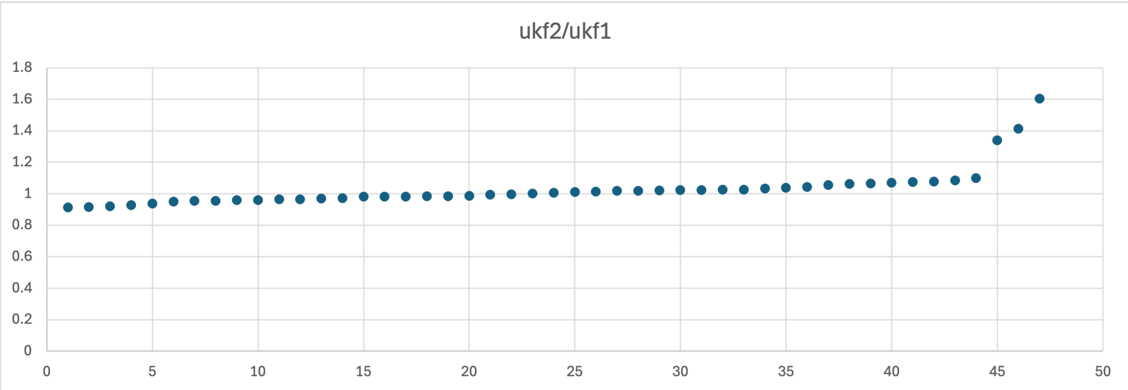


Figure 4.7: Comparing between two UKF models.

For the first 44 journeys, the energy estimates from both UKF models remain close to one another, as shown by the clustered data points around a ratio of 1. This consistency suggests that both models perform similarly well for most journeys.

However, from approximately journey 45 onward, the ratio between the two models begins to exceed 1. This increase indicates that the UKF model without control observations estimates higher energy consumption than the UKF model with control observations in these cases. This trend supports the conclusion that incorporating control observations enhances the accuracy and stability of energy estimates in the UKF model, particularly for journeys with greater variability in train control conditions.

4.6 Conclusion

We have shown that an Unscented Kalman Filter (UKF) with control observations provides similar energy estimates to the time-in-notch calculation. This is as expected—the UKF is performing the time-in-notch calculation with adjustments that consider the train dynamics.

If notch data is not available then the UKF without control observations performs reasonably well for most journeys. The simpler calculation of energy from sampled speeds is not quite as good as the UKF, but still more accurate than the current Energymiser calculation.

5 Comparing advised control to actual control

In this section we discuss preliminary work comparing Energymiser logs and locomotive logs to see how closely drivers follow Energymiser advice.

Energymiser advises four basic driving modes, indicated by a variable a:

Power ($a = 1$): the driver is expected to apply full power (Notch 8)

Hold ($0 < a < 1$): the driver is given a target speed and should vary the power and brake setting to maintain this speed as closely as possible

Coast ($a = 0$): the driver is expected to put the control into Notch 0

Brake ($a = -1$): the driver is expected to use dynamic braking, blended with air braking at low speed.

The advice value in the Energymiser logs is actually a smoothed version of the Energymiser advice—it indicates the average driving advice over the next 15 seconds. This will become clear in the following examples.

The actual driver action is indicated by three logged values: notch, air brake pressure, and dynamic brake setting. We use these values to form a control variable $u \in [-1, 1]$ as follows:

- if the dynamic brake value is greater than 0 then u is set to be the negative of the dynamic brake value
- otherwise, $u = n/8$ where n is the notch value.

5.1 Graphing the driver response to advice

We have written Python code that will search a log for points where the advice changes to Power or Coast, and plot the advised control, advised speed, actual control and actual speed for the preceeding 10 seconds and the next 120 seconds. Figures 5.1–5.3 show three examples. The top graph for each example shows the advised speed in blue and the actual speed in orange, and the bottom graph shows the advised control in blue and the actual control in orange.

In Figure 5.1 the advised control changes to Power but the driver does not respond for about 30 seconds. At about 50 seconds the advice control goes down to 0, and the driver responds about 25 seconds later. The actual speed drops below the predicted speed.

The gradual ramping up and down of the advised control (blue) is because of the 15-second averaging of the logged advice. The actual advice changes at the end of these ramps.

Figure 5.2 shows a change to Coast, but the actual speed is below the advised speed so the driver keeps applying notch 6 or 7 until the actual speed matches the advised speed, then gradually decreases the control.

In Figure 5.3 the advised control changes to Coast. Once again the actual speed is below the advised speed so the driver keeps applying power. The advice control then changes to brake, but the driver responds slowly and the actual speed increases beyond the advised speed.

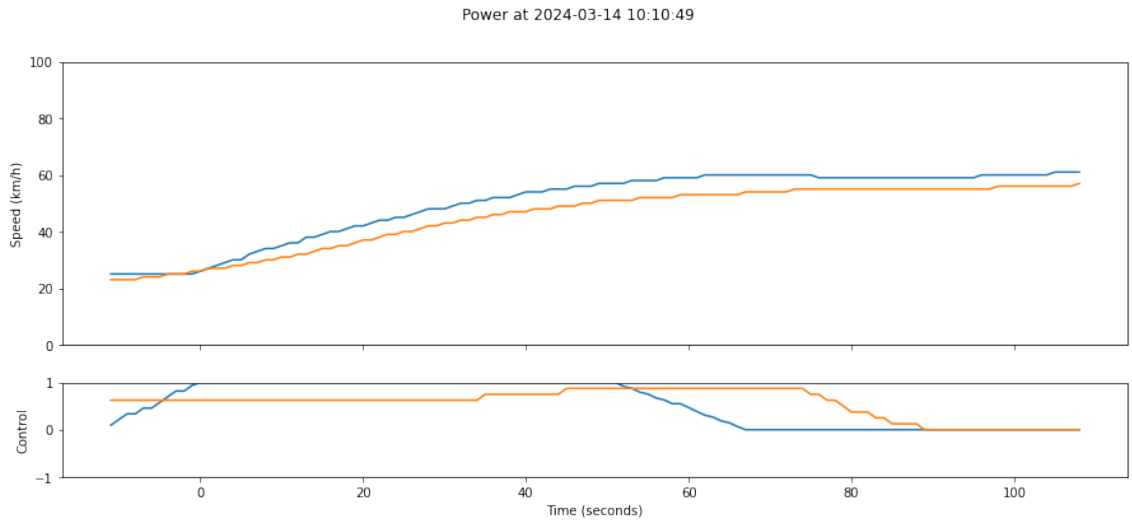


Figure 5.1: Example 1: Change to Power.

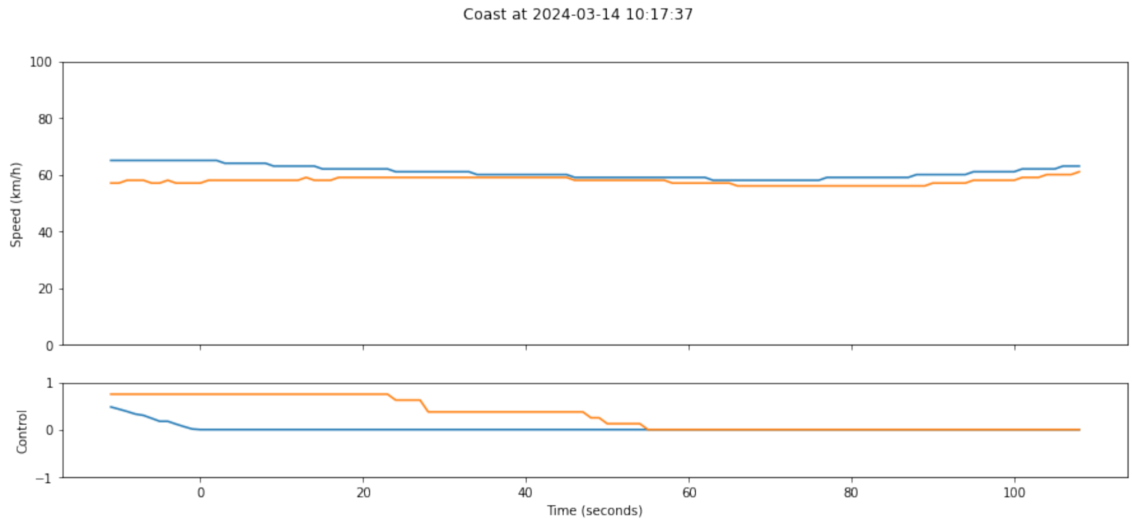


Figure 5.2: Example 2: Change to Coast.

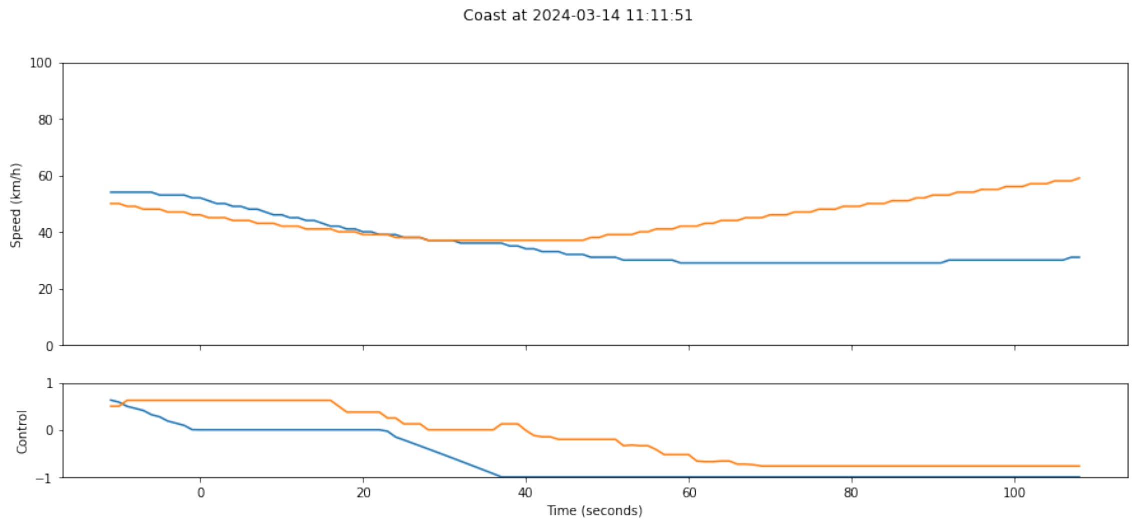


Figure 5.3: Example 3: Change to Coast.

5.2 Correlation between advice and notch

Figure 5.4 shows the correlation between advice and notch setting for an example journey. When the advised power is higher, the driver tends to select higher notches. But there is some variation. The Pearson correlation coefficient is 0.77, indicating that notch increases with advice value, as expected.

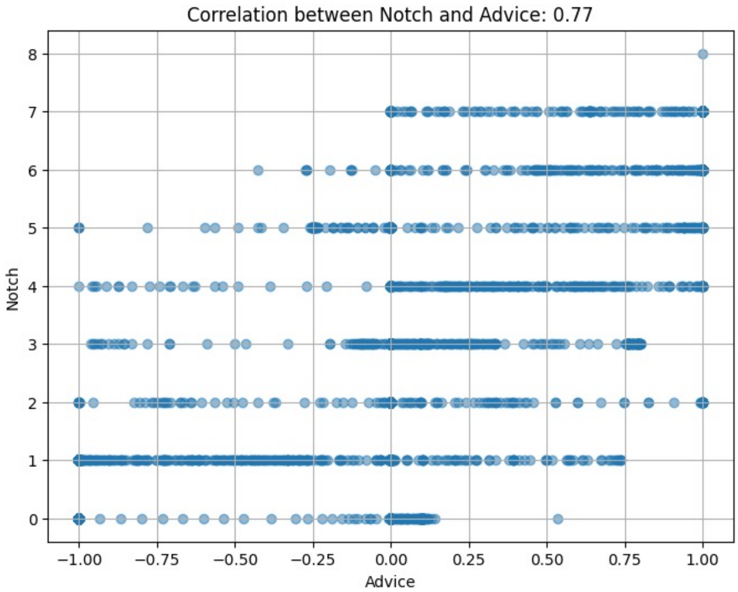


Figure 5.4: Correlation between advice and notch.

The Pearson correlation coefficient indicates the strength of a linear relationship between recommendation and control.

5.3 Control during different driving modes

Figure 5.5 shows histograms of control for an example journey when the advised driving mode is Power, Coast and Brake.

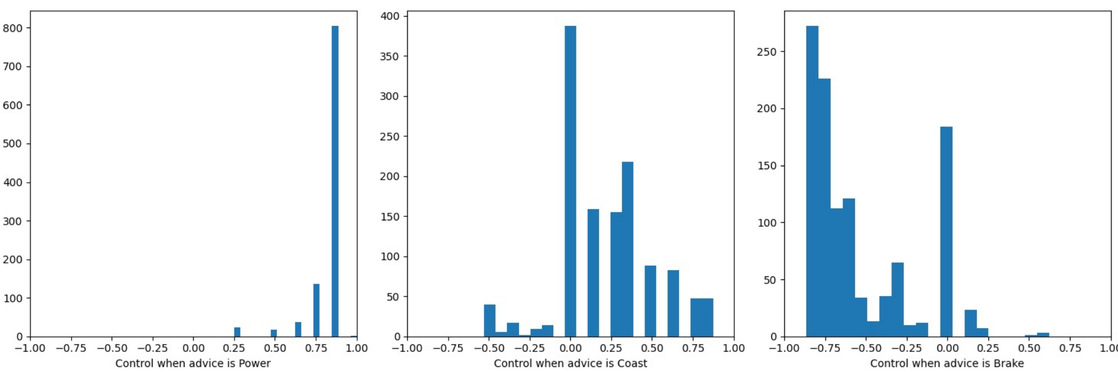


Figure 5.5: Control during Power, Coast and Brake modes.

In the first histogram, when the advised mode is Power, the driver spends most of the time at about 90% of full power, which would correspond to notch 7 rather than notch 8.

In the second histogram, when the advised mode is Coast, the driver is spending most time with zero power, but also spends time in throttle notches and braking.

In the last histogram, when advised driving mode is Brake, there is some coasting but also lots of braking.

This type of diagram may be useful for analysing individual journeys.

5.4 Correlation between control and advice

We evaluated the correlation between advice and control for each of the 50 journeys. Figure 5.7 and Figure 5.8 shows the lowest and highest correlations between control and advice for the journeys.

For the journey with the lowest correlation coefficient (Figure 5.6), the train control setting is zero for much of the journey.

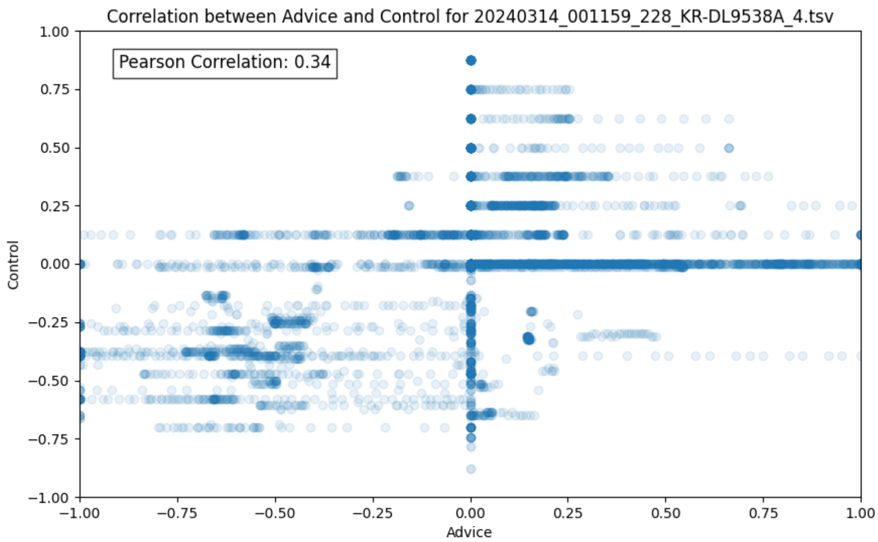


Figure 5.6: The lowest correlation between advice and notch.

If we ignore this journey, the journey with the next-lowest correlation has a coefficient of 0.39, and is shown in Figure 5.7. For this journey, the driver has ignored dynamic braking advice and used air braking instead. They have also not used notch 8.

Figure 5.8 shows the correlation between advice and control for the journey with the highest correlation coefficient of 0.92. The graph shows a clear relationship between

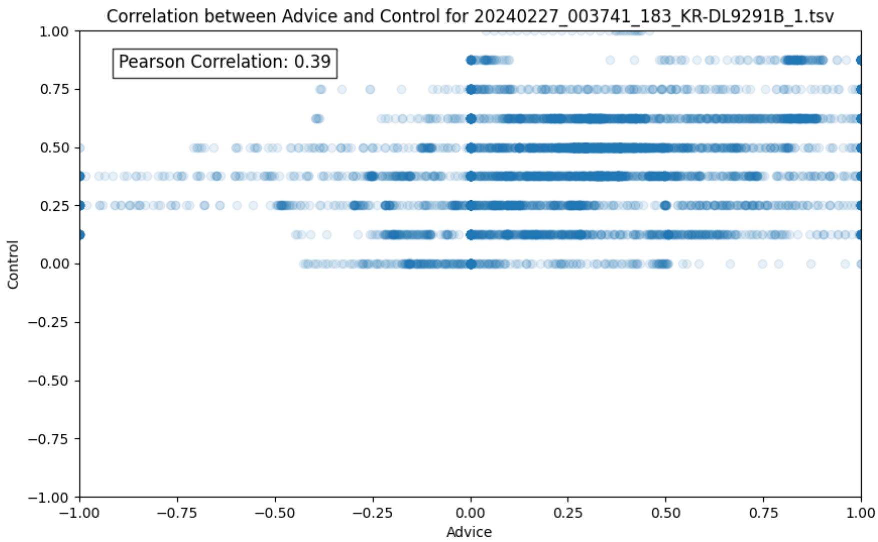


Figure 5.7: The second-lowest correlation between advice and notch.

advice on the horizontal axis and control on the vertical axis. However, there is no point when control is 1, because the driver never uses notch 8.

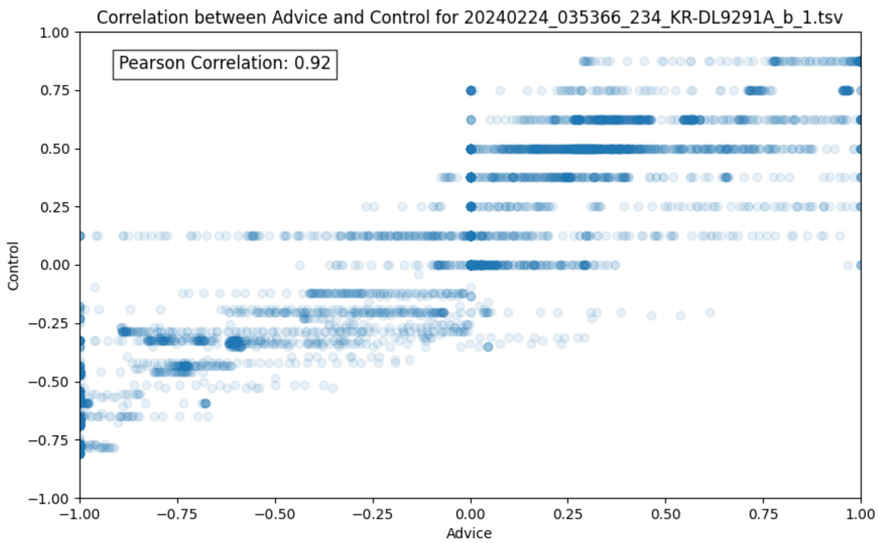


Figure 5.8: The highest correlation between advice and notch.

Figure 5.9 shows the Pearson correlation coefficients for each of 50 logs, arranged from largest to smallest. Journeys with low correlations between advice and control should be investigated further.

Table 5.1 shows the correlation for each journey, in order of highest to lowest correlation.

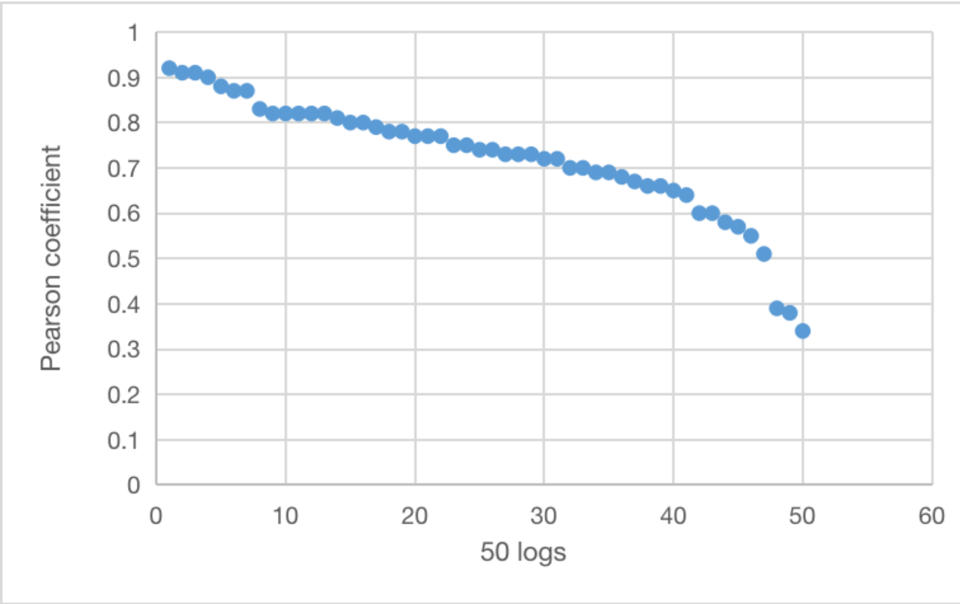


Figure 5.9: Pearson correlation coefficients for each of 50 logs.

Table 5.1: Pearson correlation coefficients for each of 50 logs.

Log	Correlation coefficient
20240224_035366_234_KR-DL9291A_b_1	0.92
20240314_014931_391_KR-DL9072B_3	0.91
20240214_125404_535_KR-DL9291B_a_1	0.91
20240314_001159_228_KR-DL9538A_2	0.9
20240213_185702_521_KR-DL9291B_1	0.88
20240228_170466_212_KR-DL9291A_b_2	0.87
20240214_125404_535_KR-DL9291B_a_2	0.87
20240201_081944_MP6_KR-DL9291A_1	0.83
20230720_004420_620_KR-DF7173_1	0.82
20240313_203920_541_KR-DL9475A_1	0.82
20230720_001308_212_KR-DL9008B_1	0.82
20240314_014931_391_KR-DL9072B_2	0.82
20240214_125404_535_KR-DL9291B_b_1	0.82
20240209_114205_MP3_KR-DL9291B_2	0.81
20240314_001159_228_KR-DL9538A_3	0.8
20240314_014931_391_KR-DL9072B_4	0.8
20240130_220634_144_KR-DL9291B_1	0.79
20240209_191900_MP10_KR-DL9291A_1	0.78
20240224_035366_234_KR-DL9291A_b_2	0.78
20240210_054351_MP1_KR-DL9291B_2	0.77

Continued on next page

Table 5.1—*Continued from previous page*

Log	Correlation coefficient
2024031_014931_391_KR-DL9072B_1	0.77
20240223_014666_566_KR-DL9291A_b_1	0.77
20230720_001308_212_KR-DL9008B_2	0.75
20240130_073151_140_KR-DL9291B_1	0.75
20240226_015821_MP9_KR-DL9291B_1	0.74
20240228_170466_212_KR-DL9291A_b_1	0.74
20240226_115932_MP8_KR-DL9291A_1	0.73
20240210_161266_MP20_KR-DL9291A_b_1	0.73
20240213_185702_521_KR-DL9291B_2	0.73
20240209_114205_MP3_KR-DL9291B_1	0.72
20240227_081322_321_KR-DL9291B_1	0.72
20240210_161266_MP20_KR-DL9291A_a_1	0.7
20240228_170466_212_KR-DL9291A_a_1	0.7
20240227_115742_MP8_KR-DL9291A_1	0.69
20240226_015821_MP9_KR-DL9291B_2	0.69
20240223_124351_565_KR-DL9291B_1	0.68
20240130_101838_141_KR-DL9291B_1	0.67
20240224_035366_234_KR-DL9291A_a_1	0.66
20240221_182358_391_KR-DL9291B_1	0.66
20240201_081944_MP6_KR-DL9291A_2	0.65
20240221_113610_390_KR-DL9291A_1	0.64
20240222_020159_328_KR-DL9291A_1	0.6
20240210_054351_MP1_KR-DL9291B_1	0.6
20240224_035366_234_KR-DL9291A_c_1	0.58
20240209_191900_MP10_KR-DL9291A_2	0.57
20240223_014666_566_KR-DL9291A_a_1	0.55
20240219_200352_215_KR-DL9291B_1	0.51
20240227_003741_183_KR-DL9291B_1	0.39
20240314_001159_228_KR-DL9538A_1	0.38
20240314_001159_228_KR-DL9538A_4	0.34

5.5 Conclusion

For journeys where control notch has been logged, the correlation between advice and control can give a good indication of how closely the advice has been followed. Journeys with low correlations should be investigated more closely—low correlation could be due to errors in the data.

6 Comparing energy use

Comparing the energy use of freight train journeys is not straightforward because different journeys have different locomotive configurations, different trailing lengths and masses, and different routes and timetables. In this section we evaluate four simple models for predicting the typical energy use of a journey. These models can be used to compare the energy use for individual journeys to the predicted energy use.

6.1 Modelling energy use

A common method for comparing energy consumption of freight journeys is to compare energy use per tonne-kilometre. However, this method ignores the length and speed of the train, which have a significant impact on aerodynamic drag.

For a train travelling at constant speed V on a level track without speed limits or stops, the force F required to overcome resistance forces has two components:

- rolling resistance, which is proportional to the mass M of the train
- aerodynamic drag, which is proportional to the length L of the train and the square of the speed, V^2.

The total work done will be $E = FX$ where X is the distance travelled.

We use this as a basis for four linear regression models:

$$\frac{E}{X} = aM + bLV^2 + c \tag{6.1}$$

$$\frac{E}{X} = aM + bLV^2 \tag{6.2}$$

$$\frac{E}{X} = aM + c \tag{6.3}$$

$$\frac{E}{X} = aM \tag{6.4}$$

where a, b and c are unknown coefficients which will be determined by fitting the models to the data. The last model, which depends on mass only, is equivalent to the common energy per tonne-km model.

Using the data from our 47 journeys gives the coefficients in Table 6.1.

Table 6.1: Coefficients for the four models

Model	a	b	c
$aM + bLV^2 + c$	0.0237	-0.0411	27224
$aM + bLV^2$	0.0373	0.0438	
$aM + c$	0.0229		23973
aM	0.0407		

6.2 Comparing journeys

We can use these models to compare the actual energy use for each journey to the predicted energy use, to see which journeys used less energy than expected and which used more. Table 6.2 shows the ratio of actual energy consumption to predicted energy consumption, evaluated using each of the four models. A value over 1 indicates that the train has used more energy than predicted by the model.

Table 6.2: Ratio of actual to predicted energy use for each journey

Log	M1	M2	M3	M4
20230720_001308_212_KR-DL9008B_1	0.95	1.11	1.00	1.07
20230720_001308_212_KR-DL9008B_2	0.71	0.76	0.71	0.77
20240130_101838_141_KR-DL9291B_1	0.73	1.44	0.78	1.45
20240201_081944_MP6_KR-DL9291A_1	0.87	0.72	0.83	0.75
20240201_081944_MP6_KR-DL9291A_2	0.88	0.82	0.88	0.81
20240209_114205_MP3_KR-DL9291B_1	0.99	0.88	0.99	0.87
20240209_114205_MP3_KR-DL9291B_2	0.89	0.78	0.88	0.77
20240209_191900_MP10_KR-DL9291A_1	0.74	0.71	0.71	0.73
20240209_191900_MP10_KR-DL9291A_2	1.19	0.98	1.07	1.09
20240210_054351_MP1_KR-DL9291B_1	1.13	1.05	1.14	1.01
20240210_054351_MP1_KR-DL9291B_2	1.00	0.93	1.02	0.90
20240210_161266_MP20_KR-DL9291A_a_1	0.95	0.97	0.95	0.97
20240210_161266_MP20_KR-DL9291A_b_1	0.79	0.77	0.77	0.78
20240213_185702_521_KR-DL9291B_1	1.77	1.79	1.79	1.76
20240213_185702_521_KR-DL9291B_2	1.35	1.33	1.35	1.33

Continued on next page

Table 6.2—*Continued from previous page*

Log	M1	M2	M3	M4
20240214_125404_535_KR-DL9291B_a_1	1.31	1.51	1.37	1.45
20240214_125404_535_KR-DL9291B_a_2	2.15	2.63	2.31	2.45
20240214_125404_535_KR-DL9291B_b_1	1.21	1.36	1.25	1.32
20240219_200352_215_KR-DL9291B_1	0.62	0.73	0.65	0.70
20240221_113610_390_KR-DL9291A_1	0.82	1.02	0.85	0.99
20240221_182358_391_KR-DL9291B_1	0.75	1.13	0.78	1.14
20240222_020159_328_KR-DL9291A_1	0.89	1.16	0.91	1.18
20240223_014666_566_KR-DL9291A_a_1	1.17	1.36	1.12	1.50
20240223_014666_566_KR-DL9291A_b_1	1.00	1.45	1.00	1.58
20240223_124351_565_KR-DL9291B_1	1.07	1.43	1.06	1.57
20240224_035366_234_KR-DL9291A_a_1	1.01	0.91	0.99	0.91
20240224_035366_234_KR-DL9291A_b_1	0.92	0.85	0.89	0.87
20240224_035366_234_KR-DL9291A_b_2	0.66	0.63	0.65	0.64
20240224_035366_234_KR-DL9291A_c_1	0.79	1.09	0.80	1.14
20240226_015821_MP9_KR-DL9291B_1	0.86	0.78	0.83	0.80
20240226_015821_MP9_KR-DL9291B_2	0.53	0.58	0.56	0.54
20240226_115932_MP8_KR-DL9291A_1	0.97	0.83	0.93	0.85
20240227_003741_183_KR-DL9291B_1	0.93	1.41	0.94	1.54
20240227_081322_321_KR-DL9291B_1	0.81	0.74	0.83	0.71
20240227_115742_MP8_KR-DL9291A_1	0.90	0.84	0.90	0.82
20240228_170466_212_KR-DL9291A_a_1	1.33	1.25	1.30	1.29
20240228_170466_212_KR-DL9291A_b_1	1.58	1.50	1.54	1.53
20240228_170466_212_KR-DL9291A_b_2	1.94	1.99	1.97	1.95
20240313_203920_541_KR-DL9475A_1	1.29	1.82	1.36	1.77
20240314_001159_228_KR-DL9538A_1	0.47	0.50	0.47	0.51
20240314_001159_228_KR-DL9538A_2	0.91	0.98	0.91	1.00
20240314_001159_228_KR-DL9538A_3	0.44	0.49	0.45	0.49
20240314_001159_228_KR-DL9538A_4	0.11	0.14	0.12	0.13
20240314_014931_391_KR-DL9072B_1	1.10	1.21	1.00	1.46
20240314_014931_391_KR-DL9072B_2	1.71	2.24	1.68	2.45
20240314_014931_391_KR-DL9072B_3	1.06	1.29	1.01	1.47
20240314_014931_391_KR-DL9072B_4	0.70	0.79	0.65	0.94

Figure 6.1 shows the relationship between actual energy consumption (horizontal axis) and predicted energy consumption (vertical axis) for each of the four models. Each blue dot represents a journey. The red dotted line indicates where the actual

energy consumption is the same as the predicted energy consumption. Blue dots below the red line indicate journeys where the actual energy consumption is higher than predicted.

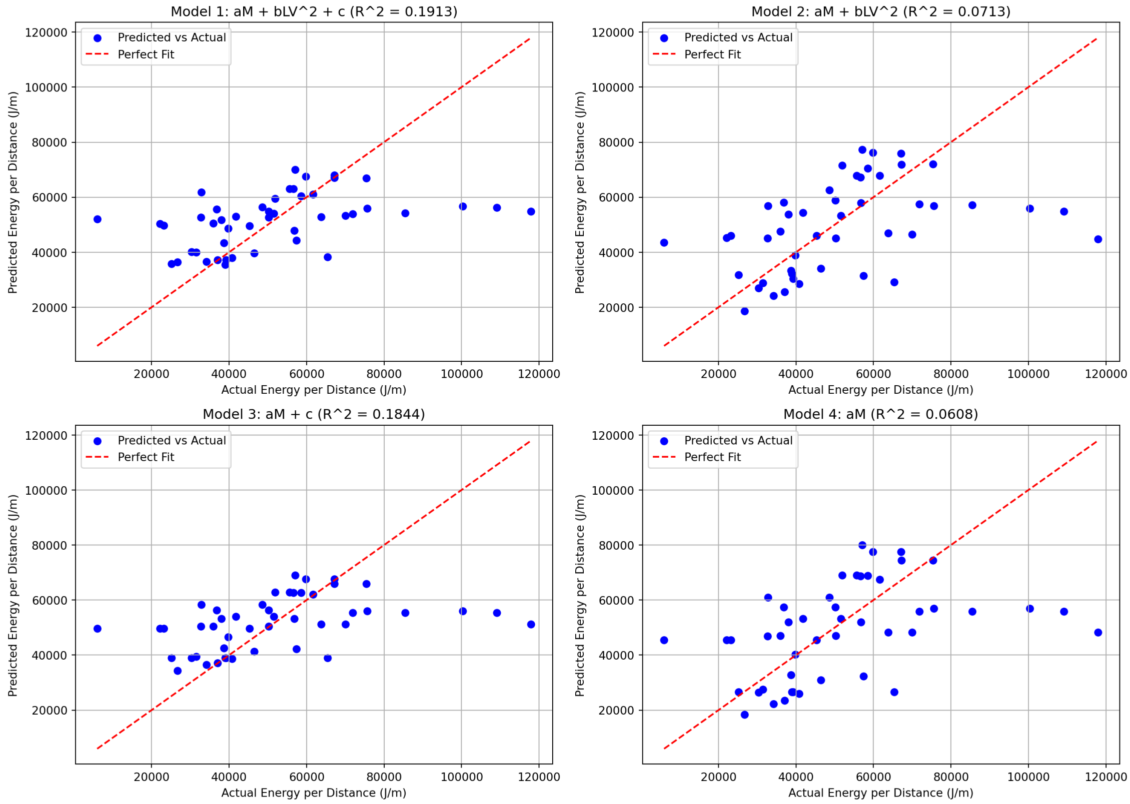


Figure 6.1: Linear regression model results

Table 6.2 and Figure 6.1 show that there are several journeys that had energy consumption significantly higher than predicted. These journeys are summarised in Table 6.3. The trains for these journeys all have length and mass significantly lower than the mean length of 541 metres and the mean mass of 1208 tonnes.

Table 6.3: Logs with high energy consumption

Log	Length (meters)	Mass (tonnes)
20240220_065630_670_KR-DL9291A	100	208
20240220_121837_671_KR-DL9291A	203	918
20240130_101838_141_KR-DL9291B	280	453
20240220_111847_671_KR-DL9291B	203	918
20240221_182358_391_KR-DL9291B	387	652
20240227_003741_183_KR-DL9291B	411	546
20240223_014666_566_KR-DL9291A_b	421	577
20240224_035366_234_KR-DL9291A_c	387	678
20240221_113610_390_KR-DL9291A	389	989

6.3 Conclusion

Comparing actual energy consumption to the energy consumption calculated from a regression model could be useful. However, like comparisons based on the correlation between driving advice and control, journeys with extreme energy consumption should be investigated to determine whether the cause of the extreme energy consumption—it could be due to unusual train configuration, unusual stopping pattern, or errors in the logged data.

7 Conclusion

The key aims of the project were to develop improved methods for estimating energy usage, and to investigate new methods for evaluating the performance of the Energymiser system.

Energymiser does not usually have access to data indicating how the train was controlled, or energy or fuel consumption data. We matched Tranzlog locomotive data from KiwiRail with Energymiser data logs to generate combined logs for each journey. By analysing speed and notch profiles of these logs, it became evident that the energy usage recorded by Energymiser is inaccurate.

We set up a simple simulation to verify that noisy speed measurements cause over-estimation of energy use, particularly when the applied tractive effort is low. We used the combined logs to show that the Energymiser system sometimes estimates significant energy consumption during periods when the train was in notch zero.

With the aim of improving energy estimates, we investigated three alternative methods for estimating energy consumption:

- If notch data is available then the **time-in-notch method** (Section 3.3) estimates energy by multiplying the time spent in each notch by the power at the wheels for that notch (which may also depend on train speed).
- The Energymiser method of calculating energy use on short intervals from the change in kinetic energy, change in potential energy and the work done against resistance can be improved significantly by using longer intervals for the calculation. Energymiser takes speed records every second and runs the energy calculation with these noisy speed records. Our **sampled-speeds method** increased the sampling interval duration to 10 seconds, and produced more accurate results.
- We developed an **Unscented Kalman Filter (UKF)** that could incorporate control observations if they were available, but otherwise estimates energy from observations of speed and track gradient. If control observations are

available then the method is essentially the same as the time-in-notch method. If control observations are not available then the method is slightly better than the sampled-speed method.

The UKF model that estimates energy from observations of speed, control and gradient force is the most accurate but also the most complicated method, and requires tuning. When notch data is availble then a time-in-notch method will give similar performance. When notch data is unavailable, the UKF model without control observations still performs reliably. The sampled-speed method, although less precise than UKF methods, is a practical alternative that provides reasonable accuracy, making it suitable when computational simplicity is needed. Further refinements to these methods could enhance their accuracy.

In addition to investigating energy usage, we looked at how train drivers responded to driving advice. If notch data is available then the correlation between advice and notch gives a good indication of how closely the driver followed the advice. Journeys with low correlation between advice and control should be investigated—the low correlation could be due to data problems.

Finally, we constructed four different regression models for predicting energy use per kilometre based on train mass, train length and train speed. This approach can be used to identify journeys that use more energy than expected.

A USB drive contents

The USB drive accompanying this report contains the following information, stored in 7 directories.

Project Brief

`Project Brief.pdf`	Project Brief given to the Mathematics Clinic group by Trapeze.

Work Statement

`Cover Letter.pdf`	Work Statement cover letter.
`Work Statement.pdf`	Work Statement.

Midyear Presentation

`Midyear Presentation.pdf`	Mid-year presentation.
`Midyear Presentation.zip`	LaTeX code for our midyear report.

Midyear Report

`Midyear Report.pdf`	Midyear report.
`Midyear Report.zip`	LaTeX code for our midyear report.

Final Presentation

`Final Presentation.pdf`	Final presentation.
`Final Presentation.zip`	LaTeX code for our final presentation.

Final Report

`Final Report.pdf`	Final Report.
`Final Report.zip`	LaTeX code for our final report.

Code

Unscented Kalman Filter

`UKF.qmd`	Quarto workbook implementing a UKF model using speed, control, and gradient force to estimate the energy use of a train.
`UKF_coasting.qmd`	Quarto workbook to exam the UKF model during a coasting phase, to demonstrate that energy use does not accumulate while coasting.

Advice Correlation

`Advice_PP.qmd`	Quarto workbook containing code to search through the log to identify instances where the advice switches to Power and Coast.
`Advice_SZ.qmd`	Quarto workbook implementing the Pearson correlation coefficient method to look at the correlation between advice and notch. It also plots histograms of the control in three driving modes.
`ControlAdvice-correlation.qmd`	Quarto workbook containing code to calculate the correlation between the advice and control for all logs.

Sampled Speed and TIN Estimation

`energy_from_speed.qmd`	Quarto workbook containing code to calculate energy from the amount of time a train spent in each notch, and estimate energy from speeds taken at 10 second intervals.

Regression Models

`regressionModels.qmd`	Quarto workbook for the creation of models comparing trains by mass, length, and velocity.

Data

`Combined Logs`	Folder containing logs combining key data from locomotive logs and Energymiser logs.
`Elevation`	Folder containing elevation data for each journey.
`DL_fuel_rates.tsv`	TSV (Tab-separated values) file containing fuel rates for each notch setting.
`regressionModelInput.txt`	TSV file containing all relevant data for producing regression models.
`trains.tsv`	TSV file containing route ID and train length, mass and resistance coefficients for each journey.
`Train_simulation_coast.xlsx`	Excel file simulating the motion of a freight train while coasting for 900 seconds.
`60 Second Journey Simulation.xlsx`	An Excel spreadsheet that simulates the motion of a train over a 60 second journey.

B Software user guide

The software we developed for the project includes Jupyter Notebooks (`.ipynb`) and Quarto workbooks with Python code (`.qmd`):

- Instructions for downloading and installing Visual Studio Code
- Instructions for downloading and installing Python.
- Instructions for downloading and installing Quarto.

The data used for the project is included on the USB drive.

Detailed documentation for each workbook is included in the workbook. The following sections give an overview of the workbooks, and what they were used for.

B.1 Sampled speed and time-in-notch estimation

The workbook `energy_from_speed.qmd` is used to calculate energy from the amount of time a train spent in each notch, and estimate energy from speeds taken at 10 second intervals.

`energy_from_speed.qmd` takes three inputs:

- `trains.tsv`, which contains train data for each journey including mass, length, resistance coefficients, and maximum power
- `Elevation`, a folder containing the elevation profile of each train route
- `combined logs`, a folder containing all journey logs.

The outputs are:

- `energy_from_speed_plots`, a folder containing a graph of estimated energy usage from each journey
- `energy_from_speed_profiles`, a folder containing files displaying the accumulated energy (in joules) from the estimation.

B.2 Unscented Kalman Filter

There are two workbooks in the Unscented Kalman Filter folder:

- `UKF_coasting.qmd` is used to implement UKF during a coasting phase. The code extracts the data from `train_simulation_coast.xlsx` to examine the UKF model during coasting. The code saves results as PDF plots.
- `UKF.qmd` implements the UKF on train data. The code extracts data from multiple sources to construct the necessary parameters for each journey. The following data sources are utilized:
  - `trains.tsv` contains the train data for each journey, including mass, length, resistance coefficients, and maximum power
  - `Data/Elevation` contains elevation information of each journey, used to calculate the gradient forces
  - `Data/combined logs` contains the combined log files for each journey.
  - `DL_fuel_rates.tsv` has power for each notch of a DL locomotive.

  The code saves the results as PDF plots.

B.3 Advice correlation

There are three workbooks in the advice correlation folder.

- `Advice_PP.qmd` is used to search through a log to identify instances where the advice switches to Power and Coast.
- `Advice_SZ.qmd` is used to calculate the Pearson correlation between advice and notch. It also plots histograms of the the control for three driving modes.
- `ControlAdvice-correlation.qmd` is used to calculate the correlation between the advice and control for all logs.

All workbooks use the data from:

- `Combined Logs`, a folder containing all journey logs.

B.4 Regression models

The workbook `regressionModels.qmd` is used to implement four linear regression models to compare journeys with different train mass, train length and speed.

`regressionModels.qmd` takes one input file:

- `regressionModelInput.txt` contains all relevant data about each journey with regards to mass, length, speed, and distance travelled.

It creates two output files, one `regressionModels.html` which includes all the code and user guide, and `cell-8-output-1.png` which is the 4 model subplots shown in Section 6.1

References

[1] Alex Cunillera, Nikola Bešinović, Niels van Oort, and Rob M.P. Goverde, Real-time train motion parameter estimation using an Unscented Kalman Filter, *Transportation Research Part C: Emerging Technologies* **143** (2022), 103794, DOI: https://doi.org/10.1016/j.trc.2022.103794.

The paper presents a method for real-time train motion parameter estimation using the Unscented Kalman Filter, tested with real data from passenger trains in the Netherlands.

[2] Phil Howlett, Peter Pudney, and Xuan Vu, Estimating train parameters with an Unscented Kalman Filter, in: *Proceedings of the Fifth Asia Pacific Industrial Engineering and Management Systems Conference*, 2004.

This paper describes the use of an Unscented Kalman Filter to estimate rolling resistance coefficients of a train coasting on a flat track.

[3] S. J. Julier and J. K. Uhlmann, Unscented filtering and nonlinear estimation, *Proceedings of the IEEE* **92** (2004), 401–422, DOI: 10.1109/JPROC.2003.823141.

This paper introduced the idea of the Unscented Kalman Filter.

[4] Roger R. Labbe, FilterPy Documentation, 2016, URL: https://filterpy.readthedocs.io/en/latest/_modules/filterpy/kalman/sigma_points.html#JulierSigmaPoints.

[5] Roger R. Labbe, FilterPy Documentation, 2016, URL: https://filterpy.readthedocs.io/en/latest/_modules/filterpy/kalman/UKF.html#UnscentedKalmanFilter.

[6] Peter Pudney, Energy-Efficient Driving for a Single Train, in: *Energy-Efficient Train Operation: A System Approach for Railway Networks*, Springer International Publishing, 2023, pp. 41–67, ISBN: 978-3-031-34656-9, DOI: 10.1007/978-3-031-34656-9_3, URL: https://doi.org/10.1007/978-3-031-34656-9_3.

This book chapter provides the dynamic equations for a train. These equations take into account various parameters such as the tractive and braking effort curves of the train, resistance forces, track gradient forces, and track curvature forces.

[7] Atul Singh, An intro to Kalman Filters for Autonomous Vehicles, 2018, URL: https://towardsdatascience.com/an-intro-to-kalman-filters-for-autonomous-vehicles-f43dd2e2004b.