

Supplerende beregninger

Forskjell ved endring av klassifisering for tunge kjøretøy

Anders V. Stubberud

11. januar 2025

Innhold

1	Intensjon	1
2	Fremgangsmåte	1
2.1	N, C, E, og ÅDTT	1
2.2	B-faktor	2
3	Resultat	2
4	Validering av resultat	3

1 Intensjon

Intensjonen er å finne verdier for trafikkbelastnings-faktorene B-faktor (etter det svenske beregningsgrunlaget) og N (hvilket avhenger av C, E, og ÅDTT, så disse er også inkludert) for to ulike klassifiseringer av tunge kjøretøy; 5.6m og 7.5m. Datagrunnlaget er hentet inn fra WIM-installasjonene på Ånestad, Øysand, Skibotn, og Verdal.

2 Fremgangsmåte

Dersom det er av interesse å se på koden som har implementert fremgangsmåten beskrevet her og generert resultatene presentert under, så er den tilgjengelig her: https://github.com/Anders-Stubberud/Engasjement-SVV/blob/main/source/features_dir/road_wear_indicators.py.

2.1 N, C, E, og ÅDTT

N er produktet av følgende faktorer:

$$N = 365 \times C \times E \times \text{ÅDTT} \times f \times \frac{(1 + 0.01p)^{20} - 1}{0.01p}$$

hvor:

- C : Gjennomsnittlig antall aksler per tunge kjøretøy. Ved å filtrere vekk kjøretøy som ikke tilfredsstiller klassifiseringen av tunge kjøretøy, ble dette snittet funnet ved å hente ut kolonnen for antall aksler per kjøretøy, summere den, og deretter dele på lengden av den.
- E : Gjennomsnittlig ekvivalensfaktor for de individuelle akslene for tunge kjøretøy. Denne beskriver bidraget til nedbrytningen av en veg sammenlignet med en 10-tonns aksel etter 4. potensregelen:

$$E = \frac{1}{n} \sum_{i=1}^n \left(\frac{a_i}{10} \right)^4$$

der a_i er vekten av aksel i . Denne faktoren ble beregnet ved å iterere over hver registrering i de filtrerte datasettene (lengde over 5.6m/7.5m), hente ut vekten av hver aksel og deretter plugge dette inn i formelen ovenfor.

- \dot{ADTT} : Gjennomsnittlig antall tunge kjøretøy per døgn i åpningsåret for vegen. Ettersom registreringene i dette datagrunnlaget ikke nødvendigvis dekker vegenes åpningsår, ble det tatt utgangspunkt i det første året med registreringer og deretter beregnet snittet av antallet tunge kjøretøy per døgn dette året.
- f : Fordelingsfaktor for tungtrafikken i kjørefeltene. Denne avhenger av antallet kjørefelt. Her er $f = 0.45$ (4-feltsveg) brukt for Ånestad, og $f = 0.5$ (2-feltsveg) for de resterende stasjonene.
- p : Forventet årlig trafikkvekst for tunge kjøretøy. Her er $p = 1$, ved antagelsen om at trafikkmengden holder seg konstant.

2.2 B-faktor

B-faktor er her beregnet som snittet av hvert tunge kjøretøys ESAL-verdi. ESAL-verdien til et kjøretøy er gitt ved:

$$ESAL = \sum_{i=1}^n \left(\frac{w_i}{10} \right)^4 \times k_i$$

hvor:

- w_i : Samlet vekt for akselgruppe i ,
- $k_i = 1$: For enkeltaksler,
- $k_i = \left(\frac{10}{18} \right)^4$: For boggiaksler,
- $k_i = \left(\frac{10}{24} \right)^4$: For trippelaksler.

B-faktoren ble beregnet ved å iterere over hver registrering i de filtrerte datasettene, vurdere aksler med mindre enn eller lik 1.8 meters avstand som tilhørende samme akselgruppe, og deretter sette dette inn i formelen ovenfor.

Merk:

I rapporten *Aksellaster og beregning av «Truck factor» fra WIM* er B-faktor definert som snittet av B_{fordon} for alle kjøretøy, hvor:

$$B_{\text{fordon}} = \sum_{i=1}^j \left(\frac{\text{axelvikt}_i}{\text{laglig last}_i} \right)^4$$

Jeg oppfatter det slik at B_{fordon} også refereres til som ESAL, men ESAL er definert som:

$$ESAL = \sum_{i=1}^n \left(\frac{w_i}{10} \right)^4 \times k_i$$

Dersom man bytter ut $axelvikt_i$ med w_i og $laglig last_i$ med 10, står man igjen med en differanse på k_i . Jeg antar at forskjellen skyldes at B_{fordon} -versjonen baserer seg på å regne på individuelle aksler mens ESAL tar akselgrupper, hvorav faktoren k_i brukes for å kompensere for dette. Isåfall vil disse fremgangsmåtene gi tilnærmet samme resultat. Beregningen av B-faktor her har tatt utgangspunkt i ESAL-definisjonen, men kan enkelt tilpasses B_{fordon} -definisjonen.

3 Resultat

Ved fremgangsmåten beskrevet ovenfor, er dette resultatet:

Sted	B-faktor 5.6	B-faktor 7.5	N 5.6	N 7.5	ADTT 5.6	ADTT 7.5	E 5.6	E 7.5	C 5.6	C 7.5
Øysand	0.14	0.23	2974905.14	2775474.87	1802.76	930.10	0.15	0.22	2.78	3.39
Verdal	0.09	0.14	1064415.13	1080272.80	1028.98	552.09	0.09	0.14	2.77	3.37
Ånestad (vestgående)	0.27	0.32	4215789.05	4198671.96	1165.70	933.20	0.25	0.27	4.06	4.57
Ånestad (østgående)	0.36	0.42	5313122.07	5223209.34	1139.75	902.70	0.33	0.36	3.95	4.42
Skibotn	0.34	0.44	1076656.13	1069230.53	229.33	154.43	0.32	0.38	3.70	4.50

Tabell 1: Forskjell ved endring av klassifisering for tunge kjøretøy

4 Validering av resultat

For å sørge for at resultatene er nogenlunde presise, så har jeg laget tester som setter forventede verdier opp mot de beregnede verdiene. Alle testene her fungerer, dog ådtt har marginale avvik mellom forventet og beregnet verdi (skylder på avrundingsfeil). Avviket er godt under 0.5%, så jeg anser dette som tolererbart.

Testene er tilgjengelige her: https://github.com/Anders-Stubberud/Engasjement-SVV/blob/main/testing/test_road_wear_indicators.py.

```
def test_aadtt():
    unique_days = randint(1, 1000)

    data = [
        STARTTIME + i * MILLISECONDS_1_DAY
        for i in range(unique_days)
        for _ in range(randint(1, 2000))
    ]

    df = pd.DataFrame(data, columns=['StartTime'])

    total_registrations_first_year = len([t for t in data if t < STARTTIME + 365 * MILLISECONDS_1_DAY])
    actual_aadtt = total_registrations_first_year / min(365, unique_days)
    calculated_aadtt = road_wear_indicators.calculate_aadtt(pl.from_pandas(df))

    assert isclose(actual_aadtt, calculated_aadtt, rel_tol=5e-3) # avrundingsfeil opp til 0.5% avvik
```

Figur 1: Test for beregning av ÅDTT. Generer et tilfeldig antall unike dager, og for hvert av disse generer et tilfeldig antall registreringer. Beregner deretter snittet av antall registreringer per dag for det første året.

```

def test_e_and_b():
    AXLE_WEIGHT = 'AxleWeight'
    AXLE_DISTANCE = 'AxleDistance'

    df = pd.DataFrame({
        f'{AXLE_WEIGHT}1': [7000, 7000, 6000],
        f'{AXLE_WEIGHT}2': [8000, 10000, 7000],
        f'{AXLE_WEIGHT}3': [9000, 10000, 8000],
        f'{AXLE_WEIGHT}4': [8000, 10000, 9000],
        f'{AXLE_WEIGHT}5': [8000, 12000, 10000],
        f'{AXLE_WEIGHT}6': [8000, 13000, 11000],
        f'{AXLE_WEIGHT}7': [None, 11000, 12000],
        f'{AXLE_WEIGHT}8': [None, 12000, 13000],
        f'{AXLE_WEIGHT}9': [None, 13000, None],
        f'{AXLE_DISTANCE}1': [0, 0, 0],
        f'{AXLE_DISTANCE}2': [4, 3, 3],
        f'{AXLE_DISTANCE}3': [1.5, 1, 1.5],
        f'{AXLE_DISTANCE}4': [6, 1, 4],
        f'{AXLE_DISTANCE}5': [1, 6, 1],
        f'{AXLE_DISTANCE}6': [1, 1, 5],
        f'{AXLE_DISTANCE}7': [None, 4, 0.8],
        f'{AXLE_DISTANCE}8': [None, 1, 0.9],
        f'{AXLE_DISTANCE}9': [None, 1, None],
    })

    weights_individual_axles = np.array([
        7, 8, 9, 8, 8, 8,
        7, 10, 10, 10, 12, 13, 11, 12, 13,
        6, 7, 8, 9, 10, 11, 12, 13,
    ])
    n_individual_axles = len(weights_individual_axles)

    weights_axle_groups = np.array([
        7, 17, 24,
        7, 30, 25, 36,
        6, 15, 19, 36,
    ])
    k_values_axle_groups = [
        1, (10/18)**4, (10/24)**4,
        1, (10/24)**4, (10/18)**4, (10/24)**4,
        1, (10/18)**4, (10/18)**4, (10/24)**4,
    ]
    n_axle_groups = len(weights_axle_groups)

    e = (1 / n_individual_axles) * np.sum((weights_individual_axles / 10)**4)
    b = (1 / n_axle_groups) * np.sum((weights_axle_groups / 10)**4 * k_values_axle_groups)

    calculated_e, calculated_b = road_wear_indicators.calculate_e_and_b(pl.from_pandas(df))

    assert e == calculated_e
    assert b == calculated_b

```

Figur 2: Test for beregning av E og B. Generer et datasett med forhåndsbestemte akselvekter og akselavstander, og beregner deretter E og B "manuelt".

```

def test_calculate_c():
    num_rows = np.random.randint(50000, 100000)
    random_axles = np.random.randint(2, 11, size=num_rows)
    df = pl.DataFrame({AXLES_COUNT: random_axles})
    expected_c = random_axles.mean()
    calculated_c = road_wear_indicators.calculate_c(df)
    assert isclose(calculated_c, expected_c, rel_tol=5e-3) # avrundingsfeil opp til 0.5% avvik

```

Figur 3: Test for beregning av C. Generer et datasett med tilfeldig antall aksler per kjøretøy, og beregner deretter snittet av antall aksler per kjøretøy.